An Architectural Model of Visual Motion Understanding

Thomas Jeremy Olson

Technical Report 305
August 1989

UNIVERSITY OF
ROCHESTER
COMPUTER SCIENCE

An Architectural Model of Visual Motion Understanding

by

Thomas Jeremy Olson

Submitted in Partial Fulfillment

of the

Requirements for the Degree

DOCTOR OF PHILOSOPHY

Supervised by Jerome A. Feldman

Department of Computer Science

University of Rochester

Rochester, New York

July 1989

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 305	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): An Architectural Model of Visual Motion Understanding	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Thomas Jeremy Olson	8. CONTRACT OR GRANT NUMBER(s): N00014-84-K-0655, DACA76-85-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Computer Science Department, 734 Computer Studies Bldg., University of Rochester, Rochester, NY 14627	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: D. Adv.Res. Proj. Agency, 1400 Wilson Blvd., Arlington, VA 22209	12. REPORT DATE: July 1989
	13. NUMBER OF PAGES: 161
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Office of Naval Research, Information Systems, Arlington, VA 22217; US Army, ETL, Fort Belvoir, VA 22060	15. SECURITY CLASS. (of this report): unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

none.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

vision, motion perception, apparent motion, connectionist models

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The past few years have seen an explosion of interest in the recovery and use of visual motion information by biological and machine vision systems. In the area of computer vision, a variety of algorithms have been developed for extracting various types of motion information from images. Neuroscientists have made great strides in understanding the flow of motion information from the retina to striate and extrastriate cortex. The psychophysics community has gone a long way toward characterizing the limits and structure of human motion processing.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

The central claim of this thesis is that many puzzling aspects of motion perception can be understood by assuming a particular architecture for the human motion processing system. The architecture consists of three functional units or subsystems. The first or low-level subsystem computes simple mathematical properties of the visual signal. It is entirely bottom-up, and prone to error when its implicit assumptions are violated. The intermediate-level subsystem combines the low-level system's output with world knowledge, segmentation information and other inputs to construct a representation of the world in terms of primitive forms and their trajectories. It is claimed to be the substrate for long-range apparent motion. The highest level of the motion system assembles intermediate-level form and motion primitives into scenarios that can be used for prediction and for matching against stored models.

The architecture described above is the result of joint work with Jerome Feldman and Nigel Goddard (Feldman, 1988). The description of the low-level system is in accord with the standard view of early motion processing, and the details of the high-level system are being worked out in (Goddard). The secondary contribution of this thesis is a detailed connectionist model of the intermediate level of the architecture. In order to compute the trajectories of primitive shapes it is necessary to design mechanisms for handling time and Gestalt grouping effects in connectionist networks. Solutions to these problems are developed and used to construct a network that interprets continuous and apparent motion stimuli in a limited domain. Simulation results show that its interpretations are in qualitative agreement with human perception.

Curriculum Vitae

Thomas J. Olson was born on December 1, 1955, in Cambridge, Massachusetts. It was probably there that he acquired the taste for student life evidenced by his later career, though his only memory of the period is seeing his father walk up the aisle to receive his doctorate in 1957. His early years were spent in Norwich, Vermont, chasing snakes and harassing his younger siblings. In 1961 he moved to Baltimore, Maryland, where he attended the public school system, made a few good friends, and developed long hair and a taste for obscure branches of knowledge. One of the latter was astrophysics, and it was with the intention of pursuing this field that he entered The Johns Hopkins University in the fall of 1973. Two years of waning interest and declining grades followed, and in his junior year he converted his major to a self-designed interdisciplinary program in Medieval history and culture.

The Middle Ages proved congenial, and he graduated in 1977 with a modest but respectable grade point average. He and his wife then moved to New York City, where he attempted to make a living performing music of the Middle Ages and Renaissance. His principle academic achievement during the period was participation in the Pan-American Congress of Shawms, a conference on Medieval double-reed instruments held at midnight in the unfinished crypt of the Cathedral of St. John the Divine.

While working a 'straight job' in a bookstore he encountered two books that were to have a major impact on his life: *The Eighth Day of Creation* by H. F. Judson, and *Gödel, Escher, Bach* by Douglas Hofstadter. The former convinced him that science was a beautiful and noble pursuit, while the latter showed him that computers had applications far beyond generating payroll checks and telephone bills. In 1980 he returned to Johns Hopkins to pursue a second Bachelor's degree, this time in Electrical Engineering. Maturity and motivation had had a remarkable effect on his mathematical ability, and he graduated in 1982 with University and Departmental honors.

He spent the next two years working for The Johns Hopkins Applied Physics Laboratory. His primary responsibility during the period was the design and programming of a signal processing computer for the TOPEX oceanographic satellite.

In 1984 academia called again, and he left APL to pursue a Ph. D. in Computer Science at the University of Rochester. His early work there, under the advisorship of Christopher M. Brown, combined an interest in vision with a talent for handling balky machines. It resulted in the publication of several papers and technical reports on vision and systems applications on the BBN Butterfly Parallel Processor. He also served as teaching assistant for the department's introductory course for new graduate students. In 1986 he developed an interest in human vision, and began working with Jerome A. Feldman on the question of how humans analyze visual motion. It was this work that led to the present thesis. In the fall of 1989 he will move to Charlottesville, VA to join the Computer Science faculty of the University of Virginia.

Acknowledgements

I would like to thank my advisor, Jerry Feldman, for innumerable contributions to my training as a scientist; but particularly for sending me to the Center for Visual Sciences at a critical moment in my student career, and thereby stimulating me to ask the questions that led to this thesis. The view of motion perception developed here came out of joint work with him and with my fellow student Nigel Goddard, and many of the ideas embodied in it are theirs.

I would also like to thank my thesis committee members, Dana Ballard, Mary Hayhoe and John Maunsell. Their encouragement kept me going through some rough stretches at the beginning, and they have been instrumental in forcing me to look hard at literature that I otherwise might have bypassed. Key ideas about coordinate systems and the representation of time have come from Mary and John, respectively. The connectionist techniques developed in chapters 5-7 owe a great deal to Dana's work, as a glance at the bibliography will show.

Chris Brown introduced me to computer vision and has been an unfailing source of advice and inspiration. His willingness to listen to half-baked ideas and read draft papers has been invaluable. I would also like to thank Carla Ellis and Tom LeBlanc for coaching my endeavors on the systems front.

I owe a great debt to the faculty and staff of the Center for Visual Sciences of the University of Rochester. Their courage in the face of the overwhelming complexity of the human visual system has been a source of unceasing amazement. I would particularly like to thank Tania Pasternak, with whom I have had many fruitful and interesting conversations, and who (with Dick Aslin) introduced me to the psychophysics and neurophysiology of motion.

My fellow students at Rochester have contributed immensely both to the work and to my life. My research would have gone far less smoothly without Ken Lynne, Nigel Goddard and Mark Fanty, who made the Rochester Connectionist Simulator the thing of beauty that it is. My conversations with them were also valuable, as were discussions with Mike Swain and Paul Cooper. Mike and Paul also helped me master the tools needed to prepare figures for the thesis. Ken Yap, Cesar Quiroz, Alan Cox and others maintained software tools on which I have relied heavily.

I would like to thank the staff of the department, who make it one of the friendliest research environments on earth; particularly Peggy Frantz, Jill Orioli Forster, and Peg Meeker, whose competence is matched only by their patience with the vagaries of graduate students.

Thanks to my parents, for understanding and support during the many twists and turns my career has taken.

Finally, I would like to thank my wife Lorraine and my children Joana and Neil. They have always consoled me when the work went poorly, and rejoiced with me when it went well. Without them this work would not have been possible.

Abstract

The past few years have seen an explosion of interest in the recovery and use of visual motion information by biological and machine vision systems. In the area of computer vision, a variety of algorithms have been developed for extracting various types of motion information from images. Neuroscientists have made great strides in understanding the flow of motion information from the retina to striate and extrastriate cortex. The psychophysics community has gone a long way toward characterizing the limits and structure of human motion processing.

The central claim of this thesis is that many puzzling aspects of motion perception can be understood by assuming a particular architecture for the human motion processing system. The architecture consists of three functional units or subsystems. The first or low-level subsystem computes simple mathematical properties of the visual signal. It is entirely bottom-up, and prone to error when its implicit assumptions are violated. The intermediate-level subsystem combines the low-level system's output with world knowledge, segmentation information and other inputs to construct a representation of the world in terms of primitive forms and their trajectories. It is claimed to be the substrate for long-range apparent motion. The highest level of the motion system assembles intermediate-level form and motion primitives into scenarios that can be used for prediction and for matching against stored models.

The architecture described above is the result of joint work with Jerome Feldman and Nigel Goddard [Feldman, 1988]. The description of the low-level system is in accord with the standard view of early motion processing, and the details of the high-level system are being worked out in [Goddard]. The secondary contribution of this thesis is a detailed connectionist model of the intermediate level of the architecture. In order to compute the trajectories of primitive shapes it is necessary to design mechanisms for handling time and Gestalt grouping effects in connectionist networks. Solutions to these problems are developed and used to construct a network that interprets continuous and apparent motion stimuli in a limited domain. Simulation results show that its interpretations are in qualitative agreement with human perception.

Table of Contents

List of Tables

List of Figures

1 Introduction

Motion and change are fundamental properties of the world we live in. We are surrounded by moving things, and our survival often depends on our ability to detect them and determine where they are going. Even in the absence of other moving things, our own head and eye movements cause the patterns of light that strike our retinas to change constantly. It is not surprising, therefore, that substantial parts of human and animal visual systems are devoted to detecting and analyzing motion.

The modern study of motion perception has its roots in the nineteenth century, but the last ten to fifteen years have seen a real explosion in our knowledge. Complementary work in psychophysics and neurophysiology has added greatly to what we know about motion perception, particularly at the lower levels. If anything, our knowledge has outstripped our understanding; we have more facts than we know how to interpret. The central goal of this thesis is to understand the human motion processing system at the architectural level, describing what the major subsystems are and what they compute. The intent is to provide a way of understanding as broad a range of phenomena as possible, and to impose an organization on the recently acquired knowledge. A secondary goal is to demonstrate that the architectural model is computationally sound and biologically plausible. Thus the work will be grounded in Computer Science, and particularly in the growing subdiscipline of connectionist modelling.

The remainder of this chapter is a condensed version of the whole thesis. The next section is a discussion of the motivation for studying motion perception in general. Section 1.2 describes the computational strategies available for interpreting motion. Section 1.3 summarizes the evidence on what strategies are used by the human visual system, and presents an outline of the architectural model developed in this thesis. The last section restates in more detail the goals and ground rules of the research.

1.1 Why Study Motion?

Visual motion is a vital source of information about the world. Its ecological relevance is obvious – moving things tend to be important. It is a good source of information about the depths of surfaces, and a powerful segmentation cue. In biological systems, it appears to be involved in a very broad range of perceptual tasks. Nakayama [1985] identifies a number of distinct uses that humans are known to make of motion information. Among them are:

detecting moving objects From an ecological point of view, moving objects are of paramount importance: they may be enemies, or food, or potential mates. It is essential to classify them quickly. Furthermore, it is essential to be able to quickly distinguish objects which really are moving relative to the current environment from objects whose retinal projections are moving due to our own head and eye movements.

recovering structure The relative rates of motion of different points in an image contain a great deal of information about the relative depths of the corresponding world points. There exist a number of demonstrations (*e.g.* [Wallach and O'Connell, 1953]) that humans do in fact obtain a vivid sense of relative depth from motion information.

recognition Work with moving light displays (MLDs) demonstrates that humans can perform complex recognition tasks based purely on motion information. Johansson [Johansson, 1973; Johansson, 1976] prepared movie sequences of actors performing simple motions (walking, jumping, etc.) in which the only light in the scene came from lights attached to the joints of the actors. In the sequences the actors themselves are not visible – only the lights can be seen. When the images are presented as a sequence of still frames, observers cannot interpret them. When they are presented as a movie, however, observers quickly identify the action being performed, can indicate which light corresponds to which joint, and can sometimes determine the sex of the actor [Kozlowski and Cutting, 1977].

proprioception Motions of the whole visual field tend to be interpreted as motions of the head and body, even when the vestibular system reports otherwise. Most of us are familiar with the falling feeling we get when, seated in a train in the station, we see an adjacent train begin to move. The effect is powerful enough [Lee and Aronson, 1974; Lee and Lishman, 1975] to cause standing infants to fall down, and to destabilize the posture of adults.

segmentation Similar motion of surface patches is a strong clue that they belong the the same physical object. People can easily segregate regions of a random-dot

field that are distinguished from their surround only by their motion [Anstis, 1970].

controlling pursuit eye movements We use motion relative to our retinas to drive eye movements that allow us to follow a moving object with our eyes. This in turn allows us to extract the maximum amount of information about that object, measure small relative motions of its parts, et cetera.

The importance of motion perception to biological vision can be demonstrated in other ways. First, consider its ubiquity. Neurons that are clearly tuned for moving patterns have been found in monkeys, rats, cats, flies, pigeons, and in fact in every higher animal examined to date. This is not true of many other properties of the visual world. For example, many animals are color blind. A second piece of evidence for the importance of motion processing is the incidence and effect of defects in various visual modules. Color blindness in humans is very common [Levine, 1985], and stereo defects are widespread as well [Marr, 1982], but persons with color or stereo defects are rarely aware of them. Motion processing deficits, on the other hand, are extremely rare.

When the motion processing system does fail, the effect can be crippling. Zihl [Zihl *et al.*, 1983] reports a remarkable case of a human who apparently suffered very specific damage to the motion centers. Her visual acuity, contrast and flicker thresholds, and many other standard measurements of visual function were completely normal, but she had great difficulty judging the direction and velocity of moving objects. This disability had a profound effect on her ability to function in everyday situations. She had difficulty pouring a cup of coffee, because she was unable to predict how soon it would overflow. She had difficulty crossing streets, because she could not determine the speeds of vehicles and plan a crossing that would avoid them. Most striking, she sometimes failed to notice major changes in the environment, such as a person approaching her, until long after they had entered her field of view.

1.2 Ways of approaching motion perception

Consider a simple image consisting of a black square moving against a white background. How might its motion be recovered? There are two fundamentally different ways to approach the problem. The approach that is simplest in terms of mechanism is based on viewing the image as a three-dimensional function of image coordinates and time. It can be shown that motion in a scene corresponds to characteristic mathematical properties of the spatiotemporal image function, and that simple non-linear filters can extract those properties. The properties can be described in various ways, though often they turn out to be mathematically equivalent. Approaches based on this strategy will be referred to here as *continuous methods*.

A second way of analyzing the motion of a square is to identify its features and track them over time. For example, one might choose to locate corners at various times during presentation of the scene and try to determine how they correspond. Such *correspondence methods* vary significantly in their general character depending on what features are tracked. In the case of the simple scene described above, choosing the square itself as a feature trivializes the matching problem, but it makes the feature extraction process more difficult.

Both continuous and correspondence methods have been studied extensively in computer vision, and will be discussed in some detail in Chapter Three.

1.3 The Structure of Human Motion Perception

Given that there are two basic approaches to recovering motion information, the obvious question for a cognitive scientist is which of them is used by the human visual system. The answer, as is so often the case in the study of perception, is that it apparently uses both. It has been known since the time of Exner [1875] that presenting flashed stimuli at wide intervals of space and time can produce a powerful impression of motion – the so-called apparent motion percept. Apparent motion can be produced by stimuli separated by more than 500 milliseconds and 10 degrees of arc. In a key experiment Braddick [1974] showed that motion percepts generated by spatially and temporally localized stimuli differ in fundamental ways from those produced by apparent motion displays. He attributed these differences to a separate short-range motion processing system. The short-range system can do things which the long-range system can not – for example, it can serve as the basis for segmentation – but it can be disrupted by relatively simple kinds of interference.

More recent work has shown that the clean picture presented in Braddick's original paper is overly simple. The spatial limit of the short-range process, which originally seemed to be fixed at 15 minutes of arc, has now been shown to be dependent on spatial frequency and potentially quite large [Chang and Julesz, 1983]. At the same time, theorists of a reductionist bent have tried to explain classical apparent motion phenomena by the same mechanisms put forward as models of Braddick's short-range process. They have had some success, but many other apparent motion effects seem to be fundamentally beyond the reach of such approaches. The situation, in short, is confused. There appear to be two processes at work, but it is not clear which psychophysical results are due to which system, or what the properties of the long-range system really are. It is this confusion more than anything else that motivated the work described in this thesis.

The evidence and arguments surrounding the short-range and long-range motion systems will be reviewed in Chapters Two and Three. They lead to a view of the architecture of the motion processing system that can be summarized as follows:

The short-range (henceforth *low-level*) system is an example of the continuous approach to motion processing. It extracts properties of the spatiotemporal contrast distribution that are consistent with translational motion. It is insensitive to many of the cues that can define form, such as color, disparity, and temporal coherence. It makes no use of knowledge about how the image is segmented, world knowledge, or other high-level information. Its output is something like a vector field representing instantaneous motions or possible motions at each point in the scene. When its assumptions are satisfied, *i.e.* when all change in the image is due to the continuous motion of objects defined by contrast patches, it works very well. When its assumptions are not satisfied, however, it frequently fails.

The character of the long-range (henceforth *intermediate-level*) system is quite different. It is a correspondence method operating on segments or shape primitives defined by some relatively sophisticated segmentation process, and its output is a description of the segment motions in terms of a primitive set of trajectories. Its computation is a relaxation process that takes into account the output of the low-level system, spatiotemporal relationships between segments, information about segment properties, world knowledge, and the observer's expectations. Because it is able to integrate many different types of information, it is much more robust than the low-level system.

In addition to the low and intermediate-level motion systems, there is a high-level system that is concerned with combinations of the primitive trajectories found by the intermediate level. It is this level that allows humans to recognize characteristic motion sequences, such as a horse's gallop. It provides a two-way link between the intermediate level's description of the moving scene and models stored in long-term memory, so that perceived motions can permit recognition and recognition can assist trajectory analysis.

Later chapters of this thesis will elaborate and justify this view of how the motion perception system is organized, and present mechanisms that might be used to implement the intermediate level of the system. The highest level will be discussed only briefly, since it is described in detail elsewhere [Goddard].

1.3.1 Motion in a larger context

Motion perception is one aspect of the more general problem of dealing with visual change. The architecture summarized above grew out of a larger effort, led by Jerome Feldman, to develop a general theory of change processing. It owes a great deal to his ideas as expressed in [Feldman, 1988] and [Feldman, 1985]. Credit is also due to Nigel Goddard, who developed the model of high-level motion perception that is assumed here [Goddard].

A key idea developed in [Feldman, 1988] is that visual perception can be viewed metaphorically (and sometimes literally) as a deconvolution problem. The signals

reaching primary visual cortex can change for a variety of reasons: the lighting may be changing, the observer may be moving, there may be moving objects present, et cetera. Often all of these will happen at once. The visual system's problem is to invert the projection process that compresses all of these changes together, recovering a description of those aspects of the change which are useful. Although it will not be stressed here, the reader should always think of the motion interpretation system as being in competition with other possible explanations of perceived changes.

1.4 Goals and Approach

The ultimate goal of this thesis is to understand human motion perception at the architectural level – that is, to be able to describe it in terms of reasonably well-characterized subsystems and their interactions. The philosophical approach is patterned after that taken by Feldman in the Four Frames model [1985]. The guiding principles are: first, to attempt to describe a very broad class of perceptual phenomena; second, to remain approximately consistent with everything known about the visual system; and finally, to describe everything at a level that could plausibly be reduced to biological hardware. This last constraint will be met by developing connectionist models [Feldman and Ballard, 1982] for any parts of the architecture whose implementation is not obvious.

The phrase 'architectural model' is intended to echo Marr's well-known distinction between computational, algorithmic and implementational models of perception. Marr argued that computational analysis should precede study at the other levels. However, it is now generally accepted that the constraints imposed by the hardware of the brain are important enough that they should be considered right from the outset (see *e.g.* [Hildreth and Koch, 1987]). Recently Tsotsos proposed that counting arguments based on hardware considerations could constrain representations even when the underlying computations are only weakly characterized [Tsotsos, 1987]. This idea, which he calls complexity-level modelling, results in a level of description similar to what is sought here. The major difference is that the present work relies more on biological data and less on pure counting arguments.

The rest of the thesis proceeds as follows: Chapter Two develops biological constraints on the model by reviewing relevant literature from the disciplines of psychology, psychophysics and neurophysiology. Chapter Three describes previous computational approaches to motion analysis, looking both at work specifically intended to model human perception and at more general computer vision approaches. The architectural model sketched above is described in greater detail in Chapter Four, and is justified on computational and biological grounds. The central conclusion of that chapter is the existence of an intermediate-level motion processing system that integrates information from many sources to arrive at a description of the world in terms of segments and trajectories. Chapter Five presents a strategy for computing

the intermediate-level representation using a connectionist network. Chapters Five and Six discuss several technical problems that must be solved in order to implement that strategy, and present candidate solutions. In Chapter Seven those solutions are used to construct a working model of the intermediate-level motion system for a restricted domain. Finally, Chapter Eight summarizes the work and its implications for vision research, and suggests directions in which it might be extended.

2 Motion in Biological Vision Systems

In this chapter we will review some of the literature on motion processing in humans and animals. The intent is not to conduct an exhaustive survey of the topic, but rather to outline the current state of knowledge in several related disciplines. This will serve to show what sort of information is available and what constraints the literature imposes on theories of the architecture of the motion processing system. Particular attention will be paid to results bearing on the question of whether there are indeed separate long and short-range processes, and if so what their characteristics are. Work on such things as recognition and kinetic depth will be omitted.

Limiting the scope of the review is not intended to limit the scope of the work. As stated in Chapter One, the goal of the thesis is to develop a model that is compatible with everything that is known about motion processing. Any exceptions to this principle will be clearly identified. Even topics that are omitted from the review, such as kinetic depth effect, are claimed to be consistent with the picture to be developed in coming chapters.

The review will be organized by discipline. Section 2.2 will review the anatomy and neurophysiology of visual cortex. The following section will discuss psychophysical results, focussing particularly on those that might apply to Braddick's proposed short-range process. Section 2.4 deals with classical apparent motion, after which we will draw conclusions about the two-systems hypothesis.

2.1 Neurophysiology

The last several years have brought major advances in our understanding of the architecture of visual cerebral cortex. The bulk of the relevant work has been done on the cat and the owl and macaque monkeys. Although the general character of the results is similar, substantial interspecies differences do exist. Unless otherwise noted, statements made in this section are based on work done on the macaque. The visual cortex can be divided on anatomical and physiological grounds into a large number of distinct regions, many of which contain maps of the entire visual field. Identification of

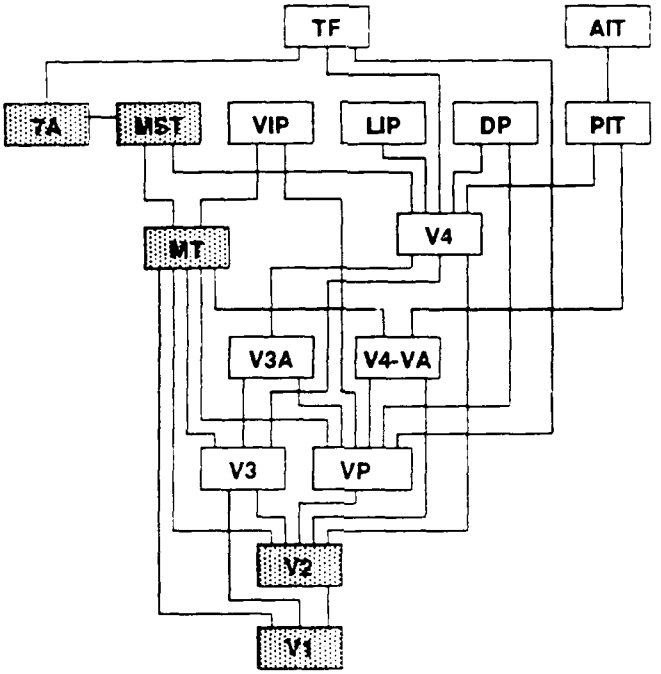

Figure 2.1: Hierarchical organization of macaque visual cortex (after [Maunsell and Newsome, 1987].) Shaded areas are those believed to play a substantial role in motion processing.

a characteristic pattern of forward and backward connections between regions suggests that in monkey cortex the regions are organized into a hierarchy like that shown in Figure 2.1 (after [Maunsell and Newsome, 1987].) Shaded areas are those believed to play a substantial role in motion processing. The various criteria used to distinguish these regions are reviewed in detail by Van Essen [1985], who also summarizes what is known about the interconnections between them.

Recent results suggest that the visual cortex contains two functionally distinct computational pathways operating in parallel. This idea was first developed in [Ungerleider and Mishkin, 1982] and is now widely accepted [Maunsell, 1987; Maunsell and Newsome, 1987]. One pathway, which leads to the temporal lobe, is concerned with color and form. Lesions of these areas tend to produce deficits in object recognition and form discrimination tasks. The other pathway, which leads to the parietal lobe, is concerned with motion and spatial relations. Both pathways enter the cortex via V1, and there are connections between them at several levels. The motion pathway leads from the retina to V1 by way of the LGN. From V1 it goes to V2 and V3, and all three of these areas feed the middle temporal visual area (MT), which is highly specialized for motion processing. Regions beyond MT that are involved in motion processing include VIP, MST and 7a, but the functions and interconnections of these areas are less well understood.

2.1.1 The Retina and LGN : Transduction and Early Filtering

In the retina, the visual image is converted to electrical potentials and subjected to some early filtering operations. Cone photoreceptors themselves have interesting temporal properties. They appear to behave something like temporal bandpass filters, with the peak response occurring at 5 Hz [Baylor, 1987]. This value coincides with the peak human sensitivity to flicker, and with the temporal frequency that produces the strongest motion aftereffect [Pantle, 1974].

Retinal ganglion and LGN cells can be divided into subpopulations with different spatial and temporal properties [Lennie, 1980]. The distinction was originally explored in the cat, though the same terminology is often used to describe similar populations in the macaque. One population, the X-type cells, is sensitive to the spatial phase of gratings, has slow-conducting axons, and tends to have a sustained temporal response. Y-type cells are insensitive to phase, have fast-conducting axons, and tend to have a transient temporal response[1]. A third class, the W-type cells, contains both sustained and transient cells and is distinguished by small size, slow response, and slightly slower axons than the X cells. In addition to X, Y and W types cats have cells that do not fit any of these classifications, including some that are directionally selective. In monkeys, however, there is no evidence for directionally selective cells prior to visual cortex.

There is good evidence that the motion/space and color/form pathways are separate even at the level of the LGN [Maunsell *et al.*, 1989], and therefor in the retina. The motion pathway seems to run from the magnocellular layers of the LGN to layer 4Cα of V1. The color pathway, by contrast, goes from the parvocellular layers of the LGN to layer 4Cβ in V1.

2.1.2 Areas V1 and V2

Motion information from the magnocellular layers of the LGN enters V1 through layer 4Cα. It is in V1 that directionally selective cells first appear. They occur in relatively large numbers in layer 4B, which is densely connected to 4Cα and projects to MT [Blasdel and Fitzpatrick, 1984; Fitzpatrick *et al.*, 1985]. In V2 motion information seems to be concentrated in "thick stripe" regions, so called for their appearance when stained for cytochrome oxidase. The thick stripes have a high incidence of direction selectivity and are insensitive to color information, relative to other regions of V2. They project to MT and V3 [Shipp and Zeki, 1985; DeYoe and Van Essen, 1985].

Foster *et al.* [1985] studied the spatiotemporal response of neurons in V1 and V2. Almost all cells in these areas exhibited bandpass spatial frequency response.

[1]Lennie [1980] points out that the sustained/transient distinction disappears in some cases, and may be an artifact of other properties of the populations such as receptive field size.

Temporal frequency response was divided between bandpass and lowpass. In V1 the majority were lowpass; in V2 the majority were bandpass, and the average bandwidths were narrower. The temporal and spatial frequency responses of neurons were for the most part separable, though this was less true for V2 than for V1. That is, the spatial frequency response of a given neuron did not vary with the temporal frequency of the stimulus. This implies that neurons in V1 and to a lesser extent V2 are not tuned for velocity in any deep sense. Similar results have been found for homologous areas in the cat [Bisti *et al.*, 1985].

2.1.3 Area MT

The middle temporal area (MT) has received a great deal of attention in the past few years. It is assumed to be deeply involved in motion processing, since a large fraction of its cells exhibit significant direction selectivity.

Receptive Field Properties

In addition to a high incidence direction selectivity, neurons in MT show receptive field properties quite different from those of V1 and V2. The classical receptive fields are large, running up to 5 degrees in the fovea and from 10 to 40 degrees in the far periphery [Mikami *et al.*, 1986]. Mikami *et al.* [1986] presented flashed dot trains at varying separations. They found directionally selective responses even when the separation between dots was larger than any receptive field in V1. This makes it seem likely that MT neurons have intrinsic direction selectivity, as well as selectivity inherited from directional subunits in lower cortical areas. That is, MT neurons do take into account the spatiotemporal arrangement of signals from their subunits[2].

Most MT neurons are selective for disparity and speed as well as direction. Thus they are well suited for analyzing motion in three-dimensional space. However, no MT neurons have yet been shown to prefer motion at different speeds in different eyes [Maunsell and Van Essen, 1983]. Thus they do not appear to have any selectivity for motion out of the frontoparallel plane.

Allman *et al.* [1985] found that many MT neurons are suppressed by presentation of their preferred stimulus outside the classical receptive field. The inhibitory region is quite large, and substantial inhibition occurs even if only a relatively small part of the surround is stimulated. This would make them well suited to play a role in motion-based segmentation.

[2]The alternative hypothesis would be that they combine information from subunits without regard to temporal order or spatial location. This would still allow them to obtain a degree of speed selectivity.

Speed Selectivity

Several studies support the idea that MT is crucially involved in extracting speed. Newsome *et al.* [1983] found that some MT neurons, unlike the V1 and V2 neurons reported by Foster (see above), do not have space-time separable tuning curves. Instead, optimal temporal frequency varies with spatial frequency in such a way as to make the neurons selective for particular ranges of velocity, independent of the spatial structure of the stimulus.

Newsome *et al.* [1985] made small lesions in MT and measured the ability of awake monkeys to initiate pursuit of moving targets. When the targets jumped to parts of the visual field corresponding to lesioned areas of MT, the monkeys' eye movements were consistent with a substantial underestimate of target speed.

Pasternak *et al.* [Pasternak *et al.*] made extensive lesions to the Lateral Suprasylvian area in cats, which is believed to be analogous to MT in the macaque. In subsequent psychophysical experiments lesioned cats showed substantial loss of the ability to discriminate speeds. However, their ability to discriminate direction of motion was comparable to that of normal cats. This suggests that that the primary areas are adequate to extract direction from the visual field, but that LS (and perhaps by extension MT) is critical for extracting velocity.

Pattern Cells

Moving stimuli whose brightness does not vary along one dimension, such as lines or gratings, are subject to an ambiguity called the *aperture problem* [Marr, 1982]. The essence of the aperture problem is that the direction of motion of a straight line cannot be determined based purely on local information. The best that can be done with local information is to determine the component of motion orthogonal to the line. Since neurons in V1 and V2 are orientation selective, they are subject to the aperture problem in the sense that they respond only to the local component of motion in a direction orthogonal to their preferred orientation. It has been demonstrated that MT has neurons that respond to the true motion of patterns rather than the motions of the pattern components [Movshon *et al.*, 1985; Albright, 1984].

2.1.4 Other Motion Areas

Far less is known about motion processing in areas beyond MT than in the areas that precede it. Saito *et al.* [1986] identified three classes of motion-sensitive cells in area MST. D-type cells were selective for to translations in the frontoparallel plane. S-type cells were selective for radial expansion or contraction of the field, and R-type cells were selective for clockwise or counterclockwise rotations or for rotations out of the frontoparallel plane. The receptive fields of these cells were very large, and the cells

responded to relatively small patches of their preferred stimulus anywhere in their receptive fields. Motter and Mountcastle [1981] describe neurons in area 7a which respond to radial flows centered on the fixation point.

2.2 Psychophysics of the Short-Range Process

Psychophysical methods have been used to explore various aspects of the visual response to moving stimuli. The methods emphasize tasks that have objective criteria, such as detection and discrimination. The very phrase 'apparent motion' implies subjectivity, so it is not surprising that much work in psychophysics has tended to avoid classical apparent motion stimuli. Good work on apparent motion has been done, but in this section we will focus on work that seems to apply more to low-level processes.

2.2.1 Spatial and Temporal Tuning

The discovery by Hubel and Wiesel of direction-selective cells in primate visual cortex motivated psychophysicists to look for perceptual correlates. Sekuler and Ganz [1963] showed that adaptation to motion in a particular direction raised contrast thresholds for detection of gratings moving in the same direction but not in the opposite direction. This, together with studies of the 'waterfall illusion' or motion aftereffect (*e.g.* [Keck *et al.*, 1976]), suggested that motion is detected by channels tuned for movement in opposite directions, and that direction and speed discriminations are made by comparing the outputs of opposing channels. Later work suggested that for the most part the channels are processed independently, *i.e.* that signals in different channels cannot be summed to improve detection [Levinson and Sekuler, 1985; Watson *et al.*, 1980]. Tolhurst [1973] used motion aftereffect to show that there are motion channels which are not involved in static vision and are tuned for lower spatial frequencies than the channels used in static vision.

Thompson [1984] used discrimination at threshold techniques to determine the bandwidths of spatiotemporal channels in the visual system. Based on his own results and those of a number of others, he concluded that there are just two temporal frequency channels, one dominating in the range from 0 to 4 Hz and the other from 4 Hz up to the critical flicker frequency. In the lower temporal frequency range there appear to be seven spatial channels, each having a bandwidth of one octave. Above four Hz there are three channels, each with a bandwidth of three octaves.

Levinson and Sekuler [1980] looked at the directional tuning of the motion channels and found it to be quite broad. However, they used moving random dot fields as stimuli in an adaptation experiment. Since such displays have motion energy in many directions, their results are somewhat difficult to interpret.

Keck *et al.* [1976] showed that raising the adapting stimulus contrast above a low minimum value did not increase the amount of adaptation. This indicates that the

affected population of neurons is carries little information about contrast. Pantle [Pantle, 1974], using sinusoidal gratings, found that adaptation was strongest when the temporal frequency at a point was 5 Hz, independent of the spatial frequency or velocity of the stimulus.

A number of results suggest that early motion processing does not have access to color information. Motion-based segmentation fails at equiluminance [Ramachandran and Gregory, 1978; Cavanagh *et al.*, 1985], and the perception of motion of sine-wave gratings is severly degraded [Cavanagh *et al.*, 1984].

2.2.2 Velocity Discrimination

McKee and various others have measured human ability to discriminate between various stimuli moving at different speeds. For smoothly moving stimuli, observers can discriminate differences as small as 3% to 5% over a broad range of speeds and spatial stimulus properties [McKee, 1981; McKee *et al.*, 1986]. Working with trains of dots in apparent and sampled motion, she found that when the displacement between successive dots is greater than 30 minutes of arc, the Weber fraction drops to an asymptotic value of about 8% [McKee and Welch, 1985]. Another result in [McKee and Welch, 1985] is that for dot stimuli speed estimates are computed very quickly. The Weber fraction usually reaches its asymptotic value of about 5% in less than 100 milliseconds.

Given that MT neurons are implicated in velocity discrimination and also show disparity selectivity (see previous section), one might expect speed discrimination to be tolerant of differences in depth. That is, speed might be measured in absolute (world) coordinates rather than retinal coordinates. However, McKee and Welch [1989] have found that humans are significantly better at comparing speeds measured in degrees per second than at comparing absolute speeds of stimuli at different distances.

2.2.3 Higher-level psychophysics

Beverly and Regan have demonstrated a number of adaptation effects that suggest the existence of channels for very specific high-level properties of the motion field. In one experiment they presented stimuli moving at different speeds to the left and right eyes, producing changes in disparity consistent with motion in depth. After prolonged exposure the depth percept faded. Cross-adaptation experiments suggest the existence of three or four channels for motion in depth [Beverly and Regan, 1973]. They were also able to produce a motion-in-depth percept by adapting to a pattern which was changing in size [Beverly and Regan, 1979]. In another experiment they gave convincing evidence of adaptation to rotations in the image plane [Regan and Beverly, 1985].

2.2.4 Segregation

The study of motion-based segregation using random dot kinematograms began with Anstis and Julesz [Anstis, 1970; Julesz, 1971], but it is best known through the influential work of Braddick. Braddick's early work used displays consisting of alternating random-dot images. The two images were uncorrelated except for a rectangular patch in the center, which was correlated but displaced by a variable distance. Under optimal conditions, observers saw an oscillating rectangle floating above a chaotically flickering field. Braddick studied the conditions under which they were able to report the orientation of the rectangle. His original finding [Braddick, 1974; Braddick, 1980] was that segregation fails if the displacement between frames exceeds about 15 minutes of arc. Segregation also requires interstimulus intervals (ISIs) to be shorter than 100 ms, and fails if the ISI contains a bright field or if the presentation is dichoptic. More recent results have changed the picture somewhat. The spatial limit varies with eccentricity, rising to 100 minutes of arc at an eccentricity of ten degrees [Baker and Braddick, 1985]. This may be due to reduced spatial resolution in the periphery, since Chang and Julesz [1983] found that low-pass filtering increased the maximum tolerable displacement. Other researchers have reported conditions under which the short-range process appears to act over at least one degree [Ramachandran and Anstis, 1983a; Petersik *et al.*, 1983].

As Braddick realized, the main significance of the segregation experiments was to suggest a fundamental difference between motion sampled at short spatial and temporal ranges and at the ranges studied in classical apparent motion. He proposed that they result from different processes, and this view is accepted by many (but not all) vision researchers. We shall return to this question after reviewing some of the literature on apparent motion.

2.3 Apparent Motion and the Long-Range Process

It was common knowledge even in the nineteenth century that a sequence of still pictures could produce the illusion of movement – various forms of mechanical kinematoscope survive to attest to this fact. The first scientific study the phenomenon was conducted by Exner [1875], who realized that it implied that motion perception was the result of a fundamental sense rather than an inference process. Systematic study of apparent motion began with Wertheimer, whose 1912 paper laid the foundations for the Gestalt school of psychology [Wertheimer, 1912]. These early studies and the philosophical issues they raised are elegantly summarized in [Kolers, 1972].

Classical apparent motion studies have a very different flavor from the psychophysical work discussed in the previous section. The Gestaltists (and later, psychologists of the perceptual inference school) were interested in the human tendency to perceive

order in their surroundings. Much of their work consists of demonstrations purporting to show the built-in cleverness of the visual system – that size change is seen as motion in depth, that ambiguous stimuli are interpreted in logically consistent ways, et cetera. They did however try to analyze systematically the principles governing interpretations, and it is this aspect of their work that has been continued by modern psychologists and psychophysicists. This type of analysis is made difficult by the malleability of apparent motion percepts. It was quite clear even to Wertheimer that attention and expectation play strong roles in determining what observers see. This creates problems with stability of criteria, suggests a need for naive observers, et cetera.

2.3.1 The Time-Distance Interaction

The most robust and widely accepted property of long-range apparent motion is the time-distance interaction, commonly referred to as Korte's Law [Korte, 1915]. Suppose two dots are presented in apparent motion. All other things being equal, the minimum asynchrony for which observers will report the perception of smooth motion will vary linearly with the spatial separation of the dots. Stimulus onset asynchrony (SOA) appears to be the critical variable, although other variables do come into play [Kolers, 1972].

A natural interpretation of the time/distance interaction is the idea that the visual system will not impute motion to a stimulus pair if the required speed exceeds some constant value. There are two problems with this. One is that the linear relationship between minimum SOA and distance invariably has a positive constant term; that is, it predicts that a non-zero SOA is required at a spatial separation of zero. The other is that the minimum SOA at any given separation varies with stimulus properties (see the discussion under "Cues", below.) A subtler interpretation of the effect would be that SOA (and the speed that it implies) affects the plausibility of motion as an interpretation for the stimulus. SOAs implying a very high speed are implausible, so the system prefers to interpret the events as unrelated.

2.3.2 Objects in Apparent Motion

Apparently any stimulus that can induce the perception of form can participate in apparent motion. Von Grünau [1979b] reported apparent motion produced by subjective contours. Apparent motion can also be obtained with figures defined only by disparity [Prazdny, 1986a] or chromatic contrast [Ramachandran and Gregory, 1978; Ramachandran *et al.*, 1973]. Prazdny [1986b] induced apparent motion between figures defined by spatiotemporal coherence — that is, between regions of unchanging random-dot pattern embedded in a field of randomly flickering dots. However, his subjects were unable to see motion between flickering regions in an unchanging field,

a curious result given that the situations are symmetrical. A possible interpretation is that the flicker may have made it difficult to interpret the regions as forms. Form information appears to be required for apparent motion [von Grünau, 1979a], and apparent motion percepts can be radically altered by small changes that induce figure-ground reversal [Ramachandran and Anstis, 1986].

A large number of researchers have reported apparent motion between stimuli of dissimilar shape and/or color. It is well known [Kolers, 1972] that good apparent motion can be obtained between circular and square dots. In such situations the shape of the stimuli appears to change smoothly during motion from one location to another [Kolers and von Grünau, 1976]. Watson [1986] suggests that apparent motion cannot be obtained between Gabor patches of dissimilar spatial frequency, a result contradicted by Baro and Levinson [1988]. Farrell and Shepard [1981] prepared stimuli in which observers could choose to see either rigid rotation through a large angle or rotation plus a varying degree of distortion through a shorter angle. If the required degree of distortion was large, the minimum SOA for rigid rotation increased linearly with angle. If the distortion was small, the minimum SOA for rigid rotation increased more rapidly than linear. They attributed this to competition from the non-rigid motion interpretation.

2.3.3 Apparent Motion Cues

A great deal has been learned by studying what apparent motion is perceived when the stimulus is ambiguous. Ambiguity always occurs when more than two spots are presented, or whenever more than one trajectory is possible. In a typical experiment, observers see a flashed spot followed by two others on either side of the original, and are asked to indicate whether the motion went to the right or the left. The clearest result is that motion to the nearer stimulus is almost always preferred [Kolers, 1972; Ullman, 1979]. There is also some tendency to favor whichever of the later two stimuli comes first, if they are not simultaneous.

Similarity of shape seems to play a minor role in resolving ambiguities. Exactly how much of a difference it makes is hard to determine, since there is no obvious metric for relative similarity of shapes. Kolers reports that motion goes to the more similar shape if no other cues are given [Kolers, 1972]. Ullman [Ullman, 1979] states that motion to stimuli of similar brightness or contrast is preferred. Chen presents evidence that topological invariants are significant [Chen, 1985].

Shepard and Zare [Shepard and Zare, 1982] found that briefly presenting a low-contrast path between two apparent motion stimulus dots strengthened the motion percept and lowered the minimum SOA for smooth motion. We have observed that paths of this type have a strong influence on the interpretation of ambiguous stimuli [Madden, 1989a].

2.3.4 Attention in Apparent Motion

As mentioned earlier, attention and bias have long been known to play a role in apparent motion. This has sometimes led the reductively inclined to dismiss the entire phenomenon as the result of suggestion. In fact, as Kolers observes, volitional control of the percept is limited. Willpower cannot overcome strong cues such as proximity, nor can it prevent the breakdown of apparent motion that occurs after prolonged repetition [Kolers, 1972]. On the other hand, if stimuli are truly balanced observers can switch between interpretations at will, as they might with a Necker cube. Informally, we have observed that using eye movements to follow one interpretation of an ambiguous stimulus tends to strengthen that interpretation, an observation also made by Burt and Sperling [1983].

Dick *et al.* [1987] have suggested that apparent motion is an attentional process in the sense of [Triesman, 1985]. They presented observers with random dot fields that underwent jumps of various sizes. In some displays a single dot moved in the opposite direction, and observers were asked to determine whether such a dot was present. When the jumps were large (25 minutes of arc), response time varied with the number of distractors; when they were small (7 minutes of arc), it was constant.

2.3.5 Coordinate Systems

Apparent motion seems to take place in a non-retinotopic coordinate system. As mentioned above, tracking one interpretation of an ambiguous apparent motion stimulus with the eyes strengthens the tendency to see that interpretation [Burt and Sperling, 1983]. Rock and Ebenholtz [1962] used artificial pupils to guarantee that the retinal positions of apparent motion stimuli remained fixed while subjects alternated gaze between two locations, and found that good apparent motion was obtained. They also showed that other cues (besides eye movements) that altered phenomenal location could induce apparent motion.

2.3.6 Apparent Motion and Short-Range Motion

A number of experiments suggest interactions between apparent motion and continuous motion. Madden [1989b] has shown in informal experiments that short-range motion applied to the first flash of an ambiguous apparent motion display biases the interpretation. Giaschi and Anstis [1989] report that apparent stimulus velocity increases with increasing ISI, which they interpret this as suggesting fusion of long-range and short-range velocity measures. Green [1983] showed that apparent motion between patches of smoothly moving sinusoidal grating was strongly affected by direction of motion of the grating. Motion in the same direction increased the range of conditions over which apparent motion was seen, while motion in the opposite

direction abolished the apparent motion, and even caused naive observers to mistake the order of appearance of the patches.

Braddick [1980] describes a case in which the short-range system appears to provide negative information. Observers view a classical bistable display called Ternus' stimulus. Two frames are alternated with a variable blank ISI. Each frame consists of three equally spaced dots or lines. The second frame consists of the first frame shifted one dot period to the right, so that the two leftmost dots in the second frame overlap the two rightmost dots in the first. When the ISI is long, observers see all three dots moving as a group. When it is short or absent, they tend to see two dots sitting still while a third dot hops back and forth across them. Braddick's interpretation is that at short ISIs the short-range system vetos any suggestion that the middle two dots are moving. This forces the long-range system to choose an interpretation involving long distance, non-rigid motion rather than the shorter rigid motion it would normally prefer.

2.3.7 Apparent Motion Paths

The short-range process is usually thought of as producing a field of velocity vectors. The long-range process, on the other hand, seems to assign trajectories to objects. An critical question for modellers is that of what kinds of trajectories can be represented.

Motion in Depth

In a now-classic paper Attneave and Block [1973] measured the minimum SOA for smooth motion between stimuli which were separated in depth as well as in retinal angle. They found that the minimum SOA was greater than would have been expected on the basis of retinal separation, but was less than predicted by a purely 3D hypothesis. They subsequently discovered that subjects' estimates of depth were highly inaccurate, and that minimum SOA varied almost linearly with *perceived* depth.

Mutch *et al.* [1983] tried to determine whether the "minimum distance" cue used for correspondence in ambiguous cases was retinal (as proposed by Ullman [1979]) or three dimensional. They found that retinal separation appeared to dominate. However, they used stimuli whose separations ranged from 16 to 40 minutes of arc. Thus it is quite possible that they were actually stimulating a short-range mechanism. Green and Odom [1986], using larger separations induced by a stereoscope, found that 3D distance determined correspondence.

Farrell [1983] and Bundesen *et al.* [1983] found that the minimum SOA for motion between objects of different sizes is consistent with the idea that the size changes are interpreted as motions in depth.

Curved Paths

Shepard and Zare [1982] (see above) used curved low-contrast paths to induce apparent motion along arcs of varying degrees of curvature and length. They found that for the most part minimum SOA for smooth motion was a clean linear function of arc length. When the paths were nearly full circles, however, longer SOAs were required. They interpreted this as evidence of competition from the shorter straight-line trajectory.

Foster [1975] showed that object shape can induce the percept of motion along a curved path. Observers were shown a stimulus consisting of a rectangle alternating between two different positions and orientations, and asked to indicate its position and orientation half-way through its motion. Their responses were consistent with motion along a curved path rather than along a straight path with rotation about the object centroid.

2.4 One Process or Two?

As we have seen, the literature on apparent motion has quite a different flavor from the lower level motion work discussed in the previous section. It is based on different experimental paradigms, methods and standards of evidence, and even tends to appear in different journals. We must ask whether the phenomena really do result from distinct visual processes, or whether, as in the tale of the blind men and the elephant, we are studying the same underlying process from two different and limited perspectives.

It is important to note that the term 'apparent motion' is not particularly well defined. Until fairly recently it was common to use the term to refer to any discrete temporal sampling of the visual field, including such things as motion pictures. It is a trivial observation that since the eye and brain are physical systems, there must be a sampling rate above which sampled and real motion are indistinguishable. Watson *et al.* [1985] give a lucid discussion and quantitative analysis of this and show that in fact such things as movies and television *should* be hard to distinguish from temporally continuous displays. No special mechanisms or high-level processes are needed to explain their appearance of veridicality.

Even below the rate at which sampling effects become detectable, a closely sampled image has much in common with the unsampled original. Because of this detectors designed for continuous motion may still work. It has been proposed that this effect is responsible for at least some long-range apparent motion phenomena [Watson and Ahumada, 1985]. The logical conclusion to this line of argument would be to assert that Braddick's distinction between long and short-range processes is incorrect– that there is only one process, which is relatively low-level in character but is available at all spatial scales. The segregation failure observed by Braddick would be interpreted

as reflecting on the nature of the segregation process rather than on motion *per se*. All other long-range motion effects would be attributed to a combination of inference and suggestion.

We find this sort of argument highly implausible. To a computer scientist interested in biology, first of all, attributing something to 'inference and suggestion' merely begs the question. The percepts in question arise in a fairly predictable way over the course of a few hundred milliseconds, and they must be computed somehow. Connectionist arguments such as the 'hundred step rule' [Feldman and Ballard, 1982] suggest that general logical reasoning processes are unlikely to be able to do the job in time. Thus the percepts may be thought of as inferences if desired, but they are inferences performed by hardware dedicated to a particular task. Surely the computations or 'inferences' performed by such dedicated hardware deserve to be thought of as a distinct perceptual process.

On a less abstract plane, note that there is an enormous body of evidence (see sections 2.2 and 2.3) supporting the idea that early motion processing is the result of simple non-linear combinations of spatial and temporal filter outputs. In the next chapter we will see that there is considerable consensus among computational modellers that early motion processing is done in just that way. However, such mechanisms would be hard pressed to handle stimuli defined by kinetic coherence, disparity, subjective contour, or chromaticity, as the apparent motion system is said to do. Data on apparent motion suggest that its input comes from form, including the notion of figure/ground segregations, rather than from raw contrast.

Another property that distinguishes apparent motion is the range of temporal interaction that it accepts. There are a number of converging lines of evidence that point to 100 milliseconds as the limit for temporal interactions in early motion perception. It is at sampling intervals of about 100 milliseconds that motion-based segregation breaks down [Braddick, 1974], and that speed discrimination Weber fractions abruptly jump [McKee and Welch, 1985]. In areas V1 and MT of macaque visual cortex, 100 milliseconds is approximately the limit for directionally selective responses evoked by pairs of flashed dots [Mikami *et al.*, 1986]. In apparent motion, on the other hand, there is no such limit. On the contrary, in keeping with Korte's third law (see previous section), at wide spatial separations apparent motion is only seen if the temporal asynchrony is substantially *greater* than 100 milliseconds [Kolers, 1972].

The interpretation that will be assumed for the rest of this thesis is the following: There are indeed two processes (at least) involved in motion perception. The earlier of the two is limited to temporal interactions of less than 100 milliseconds. Its spatial interactions tend to be short, although we will not assume any fixed limit. It is driven by the contrast signal, and its computation can be described as extracting motion-like properties from the spatiotemporal contrast distribution. The second process is active at a broader range of spatial and temporal intervals than the first, with significant

overlap. It is driven by the perception of form, so it can be activated by a wide variety of stimuli. Its computation consists of assigning trajectories to forms.

3 Computational Approaches to Motion Perception

In the previous chapter we looked at motion processing from a biological perspective. The literature reviewed was descriptive in nature, having as its goal to characterize the various ways in which motion is perceived, the limits of motion perception, and the brain structures that support it. This chapter will review work that is more analytic in nature, in which motion perception is viewed as a computational problem. The first section covers work done in computer vision. Research in this area addresses the problem in its most abstract form, without reference to biological constraints. For this reason, it provides a useful summary of the possible ways of obtaining motion information. The second section of the chapter will describe work that is specifically intended to model motion processing in humans and animals.

3.1 Computer Vision Approaches to Motion Processing

In recent years motion analysis has attracted a great deal of interest among computer vision researchers [Thompson, 1989]. The primary motivation for much of the work has been a desire to use motion information to recover the 3D structure of the world, although there has been some interest in motion segmentation as well. In this chapter we will ignore the question of how structure is recovered from motion information, focussing instead on how the information itself is represented and computed.

As stated in Chapter One, computer vision approaches to motion processing can be loosely divided into two categories: correspondence methods and continuous-function methods. The goal of continuous methods is generally to compute the optical flow field, which is the field of vectors that specifies the instantaneous velocity (in the image plane) of each point in the image. Correspondence methods can also be used to compute the flow field, but more often the correspondences are used directly.

In continuous methods the image is viewed as a continuous function of x, y, and t, and properties of the function that are related to motion are extracted. Most of these

algorithms are based on the well-known gradient relation $\partial E/\partial t = -\nabla E \cdot \vec{v}$. The gradient relation does not measure velocity per se, but rather provides a constraint on the velocity at each point in the image. (The ambiguity of the gradient relation at a point is an instance of the aperture problem). In order to compute motion, therefore, some procedure must be applied to intersect the constraints and derive a single velocity vector at each point. Methods in common use include local application of the Hough transform [Fennema and Thompson, 1979], local least-squares fitting [Kearney *et al.*, 1987], and various regularization approaches [Horn and Schunk, 1981; Hildreth, 1984].

In correspondence methods, typified by [Ullman, 1979], the image function is divided into a sequence of frames. Each frame is processed to discover features of some sort or other, and an attempt is made to find and match the same features in successive frames. The character of these algorithms depends heavily on what sort of features are used. Extremely low-level features, such as points of high variance obtained from an "interest operator", are very easy to localize but tend to be noisy and hard to match. They are typically used in support of optic flow recovery, using images taken close together in time [Bandopadhay, 1986]. High-level features such as edges or segments are easier to identify in successive images, but they can change shape, meaning that their positions can only be known approximately.

Both of the approaches to motion perception just described have characteristic strengths and weaknesses. Continuous methods have three major problems. One is the fact that real video sequences are not continuous functions, which means that partial derivatives (assuming that the gradient relation is being used) must be estimated using numerical techniques. This can be done, but introduces a certain amount of error, which may be magnified by whatever processing method is used to resolve the aperture problem [Kearney *et al.*, 1987]. Second, the methods used to resolve the aperture problem may smooth over motion discontinuities, obliterating important details. This is particularly a problem with regularization methods. Recent work with regularization in the presence of discontinuities has begun to address this, although the algorithms developed so far tend to be computationally expensive [Geman and Geman, 1984; Blake and Zisserman, 1987; Terzopoulos, 1986]. Finally, the gradient relation given above is based on a false premise: its derivation requires the assumption that the brightness of a world point's projection on the image does not change with time. As Verri and Poggio [1987] have shown, under "reasonable" assumptions about the reflectance functions of the objects in the scene this is almost never true.

Correspondence methods suffer from another set of problems. Most of their difficulties are variations on the *correspondence problem*, the problem of finding a feature located in one image in an image taken later in time. Consider first methods based on interest operators. The interest operator typically produces a list of points in the images that are in some sense distinctive, that can be expected to match relatively few points in other images. These may be useful features, such as the projections of corners of world objects, or they may be noiselike features such as shadows or spec-

ularities. Thus design of the interest operator is crucial. Because of noise, non-rigid changes in the projected shape of objects, and occlusion, not all features in one image will be found in the other. The correspondence problem in this case becomes "given m points in one image and n points in another, put some k of them in correspondence". The solution in general is to search the space of correspondences, trying to find the best match under some cost metric. It is typically computationally expensive and error prone. Methods based on matching higher-level features avoid the search problem to some degree, since they usually have far fewer features to deal with. However, higher-level features often undergo non-rigid shape changes over time, so it becomes difficult to say exactly where they are. It may become necessary to find the minimum non-rigid change needed to transform one shape into another, possibly leading to another search problem. A final problem with correspondence methods is that they tend to produce sparse results. Where continuous methods produce a vector field (which may be unreliable at some sample points), correspondence methods only produce motion estimates at features.

A system making use of both continuous and correspondence approaches to motion analysis should perform better than either approach by itself. Notice that the problems of the two approaches are non-overlapping. Continuous methods face no correspondence problem, produce dense outputs, and require a constant amount of computation per frame. Correspondence methods avoid the aperture problem and can make use of global information. Such an approach has been proposed and partially implemented by [Yuille and Grzywacz, 1988].

3.1.1 Other Methods

Some computer vision approaches to motion analysis cannot be classified as either continuous or correspondence methods. David Heeger, inspired by biological models to be discussed in the next section, used a least squares technique to estimate the distribution of energy in the spatiotemporal frequency domain [Heeger, 1987]. It can be shown that the distribution signals the direction and speed of translational motion. A related approach is that of Fleet and Jepson, who constructed filters that respond to phase change in narrow spatial frequency bands [Fleet and Jepson, 1984]. Witkin *et al.* took a very unusual approach, using scale-space techniques to find an optimally smooth distortion function mapping one image onto another [Witkin *et al.*, 1987]. The resulting distortion function can be interpreted as an optical flow map.

3.2 Biological Models of Motion Perception

In this section we will discuss previous attempts to model motion perception in humans and animals. As in Chapter Two, early motion processing and apparent motion will be treated separately.

3.2.1 Early Motion Processing

A number of models have been proposed for direction selectivity in early vision. Barlow and Levick [Barlow and Levick, 1965] proposed delayed inhibition in the null direction as a way of explaining directional selectivity in rabbit retinal ganglion cells. Torre [Torre and Poggio, 1978] and others have presented detailed neural models based on the same principle. The importance of inhibition to directional selectivity has been confirmed experimentally: disabling inhibitory connections by the local application of blocking agents causes units which had been directionally selective to begin to respond to movement in any direction (see [Hildreth and Koch, 1987] for a review).

Werner Reichardt [Reichardt, 1961] proposed a model for directional selectivity in the *Chlorophanus* beetle in which signals at one location are correlated with time-delayed signals from an adjacent location. Taking the difference between symmetric pairs of these correlators gives a signal whose strength depends on velocity. Reichardt's model has led to a whole family of motion sensors, which will here be referred to collectively as *spatiotemporal filter (ST)* models. All of the models can be viewed as taking advantage of the spatiotemporal frequency interpretation of simple translatory motion, well described in [Watson *et al.*, 1985]. The essence of the interpretation is that the spatiotemporal frequency components of a translating image all lie on a plane through the origin of frequency space. The angle between the plane normal and the f_t axis is related to the speed, and the angle between the normal and the f_x axis is related to the direction of motion. Adelson and Bergen [1985] extract the energy in regions of (f_x, f_t) space by summing the squares of quadrature pairs of spatiotemporal Gabor functions. Comparing the energy in overlapping regions allows them to compute speed in the x direction. This idea is the basis of the work by Heeger mentioned in the previous section. Watson and Ahumada [1985] propose computing the sum of two pathways which are in quadrature in both space and time. This purely linear operation gives a spatiotemporally oriented linear filter whose output oscillates at a frequency related to local velocity. Watson and Ahumada provide a detailed description of how the filters are constructed and how information about different (x, y) orientations is combined. Van Santen and Sperling [1985] show that a generalized version of the original Reichardt model is equivalent to the other two ST models.

Marr and Ullman, coming from a computer science background, proposed a model based on the gradient relation mentioned in the previous section [Marr and Ullman, 1981]. All gradient methods rely on comparing spatial and temporal derivatives to signal direction. Marr and Ullman identified the spatial and temporal derivative operators with X and Y cells in the retina and LGN, and showed that appropriate combinations of triples of these cells could be viewed as motion detectors.

Weak support for gradient models as the source of human directional selectivity comes from an experiment by Derrington and Henning [1987]. They found that brief presentations of a moving grating superimposed on a stationary one reliably lead to a

percept of motion in the wrong direction. The ST models require modification to deal with this, and it is not clear precisely how to do it. Naive gradient models appear to predict it. However, more realistic and robust gradient models would probably be constructed so as to ignore the incorrect motion.

Adelson and Bergen [1986] demonstrated that with appropriate spatiotemporal filtering[1], a variant of the gradient model can be shown to be mathematically equivalent to their ST energy model. Thus it seems that all of the current models of low-level motion processing have a common underlying mathematical structure.

The models discussed above have all been designed to agree in some way with observed characteristics of human and animal vision. Watson and Ahumada, for example, are careful to use spatial and temporal transfer functions taken from human psychophysics. All of the ST models predict such well-known illusions as the fluted square wave illusion and the reverse *phi* phenomenon. Beyond this, there have been very few attempts to relate the predictions of any of these models to neurophysiology or psychophysics. In particular, none of the models have been tested quantitatively on Braddick's discrimination task or on velocity discrimination.

3.2.2 The High-Level Process

Compared to the low-level process, the high-level process has received little attention in recent years.

Propagation Models

The early Gestalt psychologists who discovered apparent motion also proposed mechanisms to explain it [Wertheimer, 1912; Köhler, 1923]. Their work was based on an incorrect notion of how the brain is put together, but they introduced an idea that has several modern descendents. This is the idea that flashed stimuli cause a wave of activation to spread out from some starting point in the brain. The speed of the wave in the propagation medium can be made to give time/distance interactions similar to Korte's Law. This relieves the modeller of the burden of dealing with time measurement and encoding.

Farrell and Shepard [1981] propose a propagation mechanism to explain the rigid/non-rigid apparent motion tradeoff effect discussed in Chapter Three. In their model all orientations of a particular shape are represented as points on a two-dimensional manifold. A rigid transformation from one orientation to another corresponds to a path on the surface of the manifold; a non-rigid transformation corresponds to a short-cut through a higher dimensional embedding space. Unfortunately the mechanism has not been elaborated in enough detail to permit evaluation of its performance.

[1] Specifically, filtering of the type needed to prevent the misinterpretation described in [Derrington and Henning, 1987].

Waxman *et al.* [1989] propose a propagation model based on a more general theory of perceptual grouping [Waxman *et al.*, 1988]. In their grouping system, the appearance of feature tokens such as corners and edges causes activity to propagate out from the feature location by a process analogous to heat diffusion in a solid. Cycles of diffusion alternate with center-surround feedback from a separate system that detects maxima of the diffusion surface. Over time the feedback causes the detected maxima to move toward each other and fuse, which Waxman interprets as formation of a perceptual group. He suggests that such groups are interpreted as motion when the maximum due to a newly arrived feature begins to move immediately upon its appearance. In the case of apparent motion, this will happen if and only if the diffusion wave from the first stimulus has propagated out to the location of the second stimulus by the time the latter arrives.

The Waxman model gives rise to Korte's law for obvious reasons[2]. Its predictions about the range of permissible SOAs are well in line with the data of Neuhaus [1930] as reported in [Kolers, 1972]. Since it is based on features rather than contrast, it would work with stimuli defined by disparity, color, coherence et cetera. There are a number of problems, however. It does not explain how such factors as low-level motion, attention, observer expectation, stimulus shape and so on can affect the computation. It predicts pure dependence on SOA, even in the case of Ternus' stimulus where ISI plays a critical role [Petersik and Pantle, 1979; Breitmeyer and Ritter, 1986]. It does not seem capable of extracting velocity information, and the trajectories traversed by the local maxima do not bear any relation to the trajectories that are actually perceived. Finally, the representation that it produces consists of the paths traced out by the maxima during grouping. It is a little difficult to see how this information could be used to drive higher level computations.

Ullman's Minimal Mapping Theory

Ullman [1979] made one of the most serious modern attempts to explain a broad range of motion perception phenomena, particularly including apparent motion. His interpretation of the data led him to a rather different view of the phenomenon from that presented in Chapter Two.

Ullman accepted Braddick's statement that the short-range process has an absolute spatial limit of 15 minutes of arc. He then correctly observed that at somewhat larger separations, *e.g.* 30 minutes of arc, human percepts are dominated by simple, two-dimensional effects – for example, short non-rigid motions are always preferred to longer rigid motions. He concluded that the primitives of apparent motion were very low level features, such as the primal sketch tokens of [Marr, 1982]. He considered the fundamental task of apparent motion perception to be that of determining how

[2]Actually, it predicts that minimum SOA for apparent motion should rise exponentially with distance. Waxman invokes cortical scaling to linearize the relationship.

tokens seen at one time correspond to tokens seen at a later time. This contrasts with the view developed in Chapter Two. There the apparent motion system was interpreted as assigning three-dimensional trajectories to fairly high-level forms extracted by some segmentation process.

Central to Ullman's apparent motion model is the idea of affinity between tokens. Affinity is supposed to measure the likelihood that two tokens seen at different times and places in fact correspond to a single token in motion. It can also be viewed as the *a priori* probability of the motion implied by placing the two tokens in correspondence. Because it is a very general concept, affinity can be used to incorporate many of the influences on correspondence noted in the previous chapter. It would be higher for nearby tokens, for tokens with similar properties, and for tokens whose spatial and temporal separation is in accord with Korte's third law. If correspondences are as viewed as independent, then the probability of a given interpretation is just the product of the affinities of the tokens that are claimed to correspond.

Ullman used the idea of affinity to construct a cost function over correspondences whose minimum corresponds to the most probable interpretation of the scene (hence the name "minimal map theory"). He showed that a network of simple computing elements could find the global minimum by purely local operations. However, he explicitly disclaimed any interpretation of the network as an explicit model of neural computation. Following Marr, he viewed the work as a purely computational theory, unrelated to questions of implementation. In fact it is unlikely that his network formulation could be implemented directly by neurons. This is because it requires network nodes for every possible point correspondence in the visual field. This would lead to combinatorial explosion unless the spatial range of the correspondence process is strictly limited.

Ullman's view that apparent motion interpretation is a relaxation process that attempts to find a "best" global interpretation of its input agrees with that taken here. The principle points of disagreement concern what is computed (trajectories versus correspondences) and what the primitives are (forms versus primal sketch tokens.) Another, perhaps less tangible point of disagreement concerns Ullman's view of the problem as being reducible to matching between two frames. The position taken here is that motion systems should be designed to operate in a temporally continuous environment, even if they will often be exercised on discrete stimuli. In the next chapter we will begin developing a model of motion processing that is more in accord with this interpretation of the problem.

4 An Architecture for Motion Understanding

We have spent the last two chapters surveying the state of current knowledge of the human motion perception system and examining the types of models put forward to explain its behavior. In this chapter we will put this information together to develop an overall model of how the system works. The level of detail will necessarily be lower than that in some of the models discussed in Chapter Three; current knowledge simply doesn't permit a very fine-grained model of the system, particularly at the higher levels. The goal, as stated in Chapter One, is to uncover the *architecture* of the system, to understand what information is represented and (as well as possible) how it is computed.

The model will be an elaboration of the architecture sketched in Chapter One. The claim is that the motion perception system can be divided into three computational levels. The low-level part of the system computes simple mathematical properties of the visual signal. It is entirely bottom-up, and makes no use of knowledge about the world, how the image is segmented, or other high-level information. The intermediate-level process contains a representation of segments and their motions, and corresponds closely to our conscious experience of the world. It performs a relaxation computation to arrive at the best interpretation of its input at each moment in time. A large part of its input comes from the low-level motion process, but it also takes into account a variety of hard-wired "reasonableness" constraints, world knowledge, and the expectations of the observer. The highest level of the motion system assembles intermediate-level form and motion primitives (and perhaps other, non-motion events as well) into "scenarios" that can be used for prediction and for matching against stored models.

This chapter and the next are meant to be a serious attempt to describe what actually goes on in the brain. To this end the model will be stated in terms of general principles, avoiding committment to details of the model unless the data clearly support them. For example, we will argue that the visual system must represent trajectories, but will not specify what form the representation takes. The emphasis will be on identifying computational issues and suggesting approaches to the various problems that we will encounter. In later chapters we will relax these restrictions,

making assumptions as necessary to develop a model that is detailed enough to be implementable.

The next section presents an overview of the architecture. The heart of the chapter will be a discussion of the model from computational and biological perspectives, in the course of which we will flesh out the details as far as is consistent with the stated policy of avoiding unsupported assumptions. Finally, we will consider how well existing models of motion processing agree with the architecture, and use that information to motivate the more focussed work of Chapter Five.

4.1 Overview of the Architecture

We will begin by describing what the main pieces of the proposed architecture are and what sort of things they compute, deferring until later the details of how they work and how they relate to the data on human vision. Figure 4.1 shows the general structure of the architecture[1]. Perhaps its most obvious feature it is that it strongly resembles the standard computer vision view of vision as a whole. This is by design: one of the claims made here is that motion perception is not a narrow, specifically tuned sense, but rather is tightly integrated with the rest of visual perception.

4.1.1 The Low Level : Intrinsic Images

The bottom level of the architecture is a set of iconic (retinotopic) descriptions of world properties. It consists of arrays representing such things as true reflectance, texture, depth et cetera[2]. This is consistent with standard computer vision practice (*e.g.* the intrinsic images of [Barrow and Tenenbaum, 1978]) and with the known organization of extrastriate cortex into retinotopic maps of various scene features.

Most of the intrinsic images in the low-level system are not specific to motion perception. They influence motion perception indirectly, by providing information used for segmentation and matching at the intermediate level. The single exception is the plane labelled 'retinal slip' in figure 4.1. This plane computes an approximation to the optic flow field. That is, it assigns to every point in the image a vector describing the instantaneous local motion. It is meant to be identified with Braddick's short-range process, and is the substrate for motion aftereffect, for detection of moving gratings, and for motion segmentation.

[1]The architecture presented here owes even more debt than usual to my co-workers Jerry Feldman and Nigel Goddard. It represents my perspective on the more general story found in [Feldman, 1988].

[2]It is quite possible that some constancies are actually computed and stored at the intermediate level, which is in a non-retinotopic frame. That would be consistent with [Feldman, 1985], on which the present model is based. Which alternative is preferred is irrelevant to the model at hand. The model does however require that short-range motion information be available in a retinotopic frame.

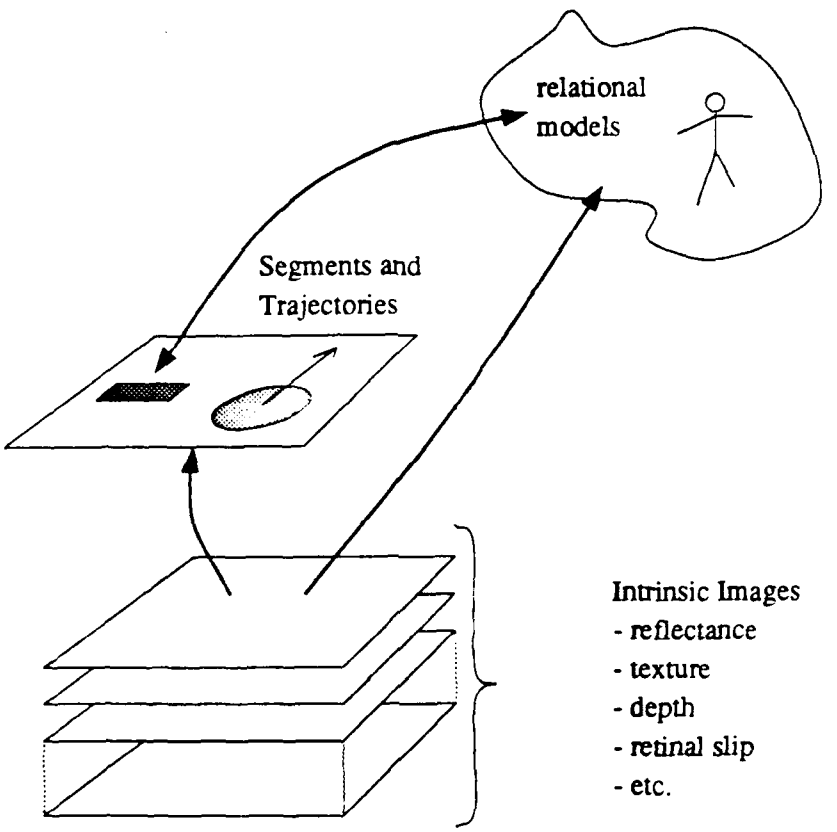

Figure 4.1: Overview of the architecture

Computationally, the low-level process is best thought of as extracting properties of the spatio-temporal contrast distribution that are consistent with translational motion. As stated in Chapter Three, there are a number of more or less equivalent ways of describing these properties: as energy distributions in ST frequency space, as ratios of filtered derivative images, et cetera.

4.1.2 The Intermediate Level : Segments and Trajectories

At the intermediate level it is assumed that the visual system uses the feature maps of the low level to partition the scene into segments or other shape primitives. The segmentation is based on the intrinsic images computed by the low level, including retinal slip.

The role of the motion-specific part of the system at this level is to assign trajectories from some primitive set to the segments. The process is a relaxation that combines input from the low-level system with other information to arrive at a 'best' interpretation of the scene. The set of primitive trajectories is unclear but includes at least straight lines and arcs at constant speeds. Unlike the low level system, the intermediate level representation is non-retinotopic. We will say more about possible coordinate systems later on, but for now let us assume a head-centered frame in the style of Feldman [1985].

The intermediate level motion system is responsible for a large part of the phenomenology of motion perception. In particular, it is the substrate for long-range apparent motion.

4.1.3 The Highest Level : Relational Models

The highest level of the motion system relates motion information to categorical knowledge. It is this level which allows humans to connect motions of segments along trajectories with things that they know about – for example, to identify a cluster of moving, brown cylinders as a horse, and furthermore to say whether it is galloping, trotting, cantering or whatever. The representation here is not iconic, though it presumably includes geometric information. Following the Four Frames model (and standard computer vision practice), objects at this level will be represented by graphs describing what the object parts are and how they are related.

Motion information at this level will be represented by sets of what amount to finite automata associated with object models. The automata (or *scenario engines*) describe complex motions in terms of sequences of primitive trajectories executed by object parts in an object-relative coordinate system. This idea is based on work by Goddard [Goddard], who describes ways of recognizing objects based on part motions from information like that provided by the intermediate level.

The most obvious perceptual phenomenon that high-level motion processing would support is recognition of moving light displays. As Johansson showed [Johansson, 1973], humans can recognize other humans engaged in characteristic activities solely on the basis of motion information. On reflection, however, it seems obvious that such a faculty is necessary even in much less contrived situations. Characteristic motions and motion sequences are a strong cue to the intent and condition of other agents. Humans have a remarkably sophisticated ability to recover that information and make inferences from it. Their obvious ability to distinguish between running, skipping and walking is impressive enough. Consider however the range of fine distinctions that can be drawn between different types of walks: swaggering, sauntering, striding, limping, et cetera.

4.1.4 Interactions

The three stages of the model interact in various ways. The primary information flow is from lower to higher levels. The low-level system provides the information needed for segmentation, and constrains the set of possible trajectories via retinal slip and property match information. The intermediate level in turn supplies input to the high level in the form of a sequence of events that can be matched against stored scenarios.

Information also flows from higher to lower levels. A higher-level scenario that matches the input strengthens the intermediate-level trajectories that support it, thereby aiding the intermediate level relaxation. Intermediate level information is used to group low-level measurements over space and time. When a detailed estimate of segment speed is required, for example, retinal slip units along the associated trajectory are pooled to form the best possible estimate. The model does not require it, but segmentation information may also help solve the aperture problem by defining which low-level unit responses are to be combined.

4.2 Why the Architecture Makes Sense

Having described the architecture at a coarse level of detail, we will now turn to the question of how well it fits computational and biological constraints, filling in some of the details along the way. It will be argued that the architecture makes good sense both in terms of what is known about human vision and in terms of what is known about motion computation in general. The computational argument will necessarily be qualitative, since the architecture itself is so loosely specified. On the biological side, we will show that the model provides natural explanations for a great variety of psychophysical results. Most of the argument will concern the low and intermediate levels, since the high level is discussed in detail elsewhere [Goddard; Feldman, 1988].

4.2.1 The Computational Argument

One of the most obvious characteristics of the architecture is that it strongly resembles the current consensus in computer vision about how a full-function vision system should look. Every serious attempt to handle vision up to the level of recognition has subsystems to handle early vision, segmentation and model matching. The details, of course, are hotly debated. At the model level, one must choose between fully three-dimensional models and characteristic or principal views; one must choose what properties to store in the models; and one must find efficient matching algorithms. At the intermediate level there are many families of shape primitives to choose from : generalized cylinders [Binford, 1971], "geons" [Biederman, 1987], and warped superquadrics [Pentland, 1986], to name only a few. At the low level there is the question of what properties should be computed, and how, and how accurately. Another problem is that of how to compute the intermediate-level shape descriptors given the low-level feature maps. The proposed architecture is compatible with most of the alternatives, though, simply because it is so qualitative.

From a computational point of view, the major novel feature of the architecture is the inclusion of trajectory information at the intermediate level. Previous attempts to compute trajectories have generally been part of very domain-specific systems. There do not appear to be any other general vision architectures that incorporate trajectory analysis as part of routine visual processing.

Why intermediate level motion?

Although nothing has yet been said about how trajectories should be recovered, it is possible to make a fairly substantial argument that trying to do so makes good computational sense. Doing so requires a null hypothesis with which to contrast the idea; let us assume that the alternative is to simply copy retinal slip information from the low level to the intermediate level, taking into account the shift from retinal to intermediate-level coordinate systems.

The most obvious argument in favor of trajectories is simply that they encode ecologically useful information. They allow the motions of objects to be predicted far enough into the future to be relevant to motor planning, permitting us to catch a baseball or dodge a flung tomato. Goddard's work shows that they can also be used to index into a model base for recognition. The utility of a detailed optical flow map is less clear. A very accurate flow map can be used to compute depth information, but attempts to do so have run into numerous difficulties, most often due to the inherent numerical instability of the problem. Pentland [1986] has suggested that constructing detailed property and depth maps is not particularly useful in and of itself, and that the proper goal of early vision should be to subserve segmentation into useful primitive shapes. Assigning primitive trajectories to segments can be thought of as an extension of this idea into the time domain. One of the claims of this work is that the primary

use of retinal slip is to aid in the recovery of trajectories, and that the low-level representation should be no more detailed than needed to support that. (Having said that, it must be admitted that humans unquestionably do use retinal slip to get depth information. The claim made here is that the depth information so obtained is qualitative, unreliable, and in general only of use in constructing intermediate-level models of the scene.)

Another problem with the null hypothesis concerns temporal invariance. Retinal slip or optical flow vectors are (ideally at least) instantaneous properties of the scene, and as such are continuously changing. This makes them hard to use for indexing into a model base. A trajectory representation would, for the same input, be stable over a longer period of time, simply because it describes properties of the scene which are true for longer intervals.

Perhaps the most serious strike against optical flow as the primary precategorical motion representation is the difficulty of computing it. Over the last ten years many very capable computer vision researchers have made serious attempts to recover optic flow, without (it must be said) great success. This is not terribly surprising, as the problem is inherently very difficult. Consider the problems faced by a unit that is trying to compute the local instantaneous flow vector based on a record of the local spatial contrast over an interval of a hundred milliseconds or so. If there *is* no spatial contrast, the unit can conclude nothing – it has no information. To recover the local flow, therefor, it will have to make inferences based on measurements taken at neighboring points. This will involve making assumptions about surface shape, orientation and continuity. Even if a given surface patch does have useful spatial contrast, there will still be ambiguity if the contrast is invariant in some direction (*i.e.* if the sample is subject to the aperture problem.)

The sparse or ambiguous local measurement problem just cited makes the optic-flow recovery problem ill-posed. Current computer vision approaches to optic flow often resort to sophisticated strategies for regularizing the problem, often including segmentation as part of their processing. That is, they optimize a functional whose arguments are the input data, the constructed flow map *and* a segmentation. Examples include [Murray and Buxton, 1987; Koch *et al.*, 1989; Koch *et al.*, 1986] and others. Unfortunately, even these strategies are insufficient to permit robust recovery. The problem is that even where a surface patch has an unambiguous time-varying contrast distribution, making a judgment of the local flow vector requires assumptions about the *cause* of the apparent change. Most formulations assume that all change is due to motion, which has been shown to be untrue in many imaging situations [Verri and Poggio, 1987]. Recognition of this fact has led some researchers to explore alternative constraints, but the solutions found to date have their own problems, such as requiring accurate measurement of second derivatives.

The above argument is purely qualitative, but it suggests that computing an accurate optical flow map is not a realistic goal. This is not to say that low-level

information is not useful – it is, and it should be used when it is available. However, we should not expect it to be available all the time, nor should we have unreasonable expectations about its accuracy. This leads back to the need for an intermediate level motion system. Given that low-level motion information will be noisy at best and useless at worst, some way is needed to supplement it with information from other sources. One obvious approach would be to try to integrate measurements over time. Recent work on shape from motion [Ullman, 1984; Grzywacz and Hildreth, 1987; Matthies *et al.*, 1987] has recognized the usefulness of this approach. A moving object gives rise to a patch of flow vectors that changes position with time, so in order to make use them one must first realize that they belong together – in other words, one needs to know the trajectory of the object.

Time averaging is not the only way to handle the unreliability of low-level motion estimation. Another source of information is the matching of shape primitives over time. That is, given a series of segmented scenes, one might try to establish the correspondence between segments in successive views. This would require comparing properties of segments across time and weighing the plausibility of correspondence given the match quality and the reasonableness of the inferred motion. The correspondence need not necessarily involve a notion of trajectory. However, trajectories provide a principled way of interpreting changes in the properties of a segment over time. For example, a size difference between two segments might indicate a correspondence error, but it might also indicate a trajectory in depth. Likewise, a change in orientation might suggest motion along a curved trajectory.

A final way to tolerate low-level motion errors is to incorporate information from the highest level. If a set of segments can be identified as a running man, for example, it should be possible to use knowledge of human locomotion to infer what his legs are doing. It seems unlikely that any reasonable high-level model would store such knowledge as flow vectors; a crude parametric description of the trajectories of parts over time is much more plausible. Thus high-level information is likely to be available in a form compatible with intermediate-level primitives rather than low-level primitives.

4.2.2 The Biological Argument

Having argued that there are good computational reasons for the computational architecture described above, we will now present evidence that human motion perception is in fact organized in this way. The claim is that the proposed architecture gives a better account of a broad range of phenomena than any competing model. As in the previous section, the principle focus will be on the low and intermediate levels of the system.

Table 4.1 summarizes the major characteristics of the low and intermediate levels and suggests how a number of well-known perceptual phenomena map to the architecture. In the paragraphs that follow we will state the case for assigning these and

Property	Low-Level System	Intermediate-Level System
stimulus	contrast	anything that can induce a percept of form
speed resolution	2–5%	8–10%
max Δt	< 100 ms	750–1000 ms
coordinates	retinal	non-retinal
phenomena	Braddick's short-range process, motion after-effect (MAE), simple direction selectivity, pursuit tracking error, proprioception	long-range apparent motion, motion phenomenology, "non-fourier motion"

Table 4.1: Characteristics of the low and intermediate level motion systems.

other phenomena to one system or the other. In many cases the assignment is an oversimplification; many effects arise because of interactions between the two systems. The discussion will also provide an opportunity to explain in more detail how the low and intermediate level systems work.

Short-Range Apparent Motion

Not surprisingly, the low-level system is identified with Braddick's short-range process [Braddick, 1974]. The low-level system can drive segmentation, like the short-range process. The intermediate-level system's inputs are segments, so an unsegmented stimulus does not activate it. Like the short-range process, the low-level system responds only to contrast stimuli over short temporal intervals.

Motion Aftereffect (MAE)

It is proposed that classic demonstrations of motion aftereffect, a.k.a the waterfall illusion, are due to fatigue in the low-level system. Note that MAE is usually described as retintopic and sensitive only to contrast signals. Of course, the neural substrate of the intermediate level system can be fatigued as well. One would expect low-level stimuli to fatigue both systems, while intermediate-level stimuli should fatigue only the intermediate level. This is consistent with results of cross-adaptation experiments with long and short-range motion [Anstis and Giaschi, 1985].

Speed Discrimination

McKee has demonstrated that the speeds of contrast stimuli in continuous motion can be discriminated with an accuracy of better than 5%. Accuracy drops to around 10% when the stimuli are sampled at frequencies falling outside the 100-millisecond

interaction time of the low-level system. Madden [1989b] found the Weber fraction for speed in apparent motion to be about 10% over a broad range of spatial and temporal separations. The Weber fraction for speed with 'non-fourier' stimuli may also be around 10% (Suzanne McKee, personal communication), though Turano and Pantle [1989] recently obtained lower figures. It is proposed that the intermediate level system can measure speeds with an accuracy of about 10% using its built-in directionally selective mechanisms. When the observer requires a finer judgment of speed, the intermediate level representation is used to select a population of low-level units on which to base that judgment. A recent result by McKee and Nakayama [1988] supports this idea. They found that when the trajectory of a test stimulus is continuous with that of a stimulus moving at a fixed reference speed, the accuracy of speed judgments drops dramatically. Providing a cue to make the reference and test trajectories distinguishable restores the accuracy to its normal level. McKee's proposed explanation is compatible with the architecture proposed here.

The phenomenology of speed perception supports the architecture in an interesting way. Naive observers in speed discrimination experiments have a strong tendency to confuse contrast and speed [McKee, 1981; McKee *et al.*, 1986]. That is, low-contrast stimuli appear to be moving faster than they really are. For a given stimulus, contrast is actually a rather good indicator of speed, because contrast sensitivity decreases sharply with increasing temporal frequency in the frequency ranges of interest. However, there is good evidence that low-level motion processing is insensitive to contrast magnitude [Keck *et al.*, 1976]. It is suggested that contrast information is factored in at the intermediate level, as part of the relaxation by which trajectories and speeds are assigned to the stimulus. When observers are presented with random-contrast stimuli they require a long training period to unlearn the correlation between contrast and speed.

Coordinate Systems

There is hardly any question that early motion processing is retinotopic. Observe that motion after-effect is retinotopic, that eye movements interfere with fine speed judgments, and that motion-sensitive cells in much of striate and extrastriate cortex are retinotopically organized. It is less clear what coordinate system the intermediate-level segments and trajectories are represented in. Feldman argues [Feldman, 1985] that any frame used for indexing must be stable over eye movements, and proposes a head-centered frame. To this one can add that motion phenomenology, which in the model is a product of the intermediate level system, is very definitely not retinotopic. In addition to the apparent motion results presented in Chapter Two, two common experiences demonstrate this. First, attempting to fixate a perifoveal afterimage produces a very compelling impression that the afterimage is moving, even though it is actually printed on the retina. Second, note that during a pursuit eye movement the target remains fixed on the retina while the background drifts. The percept, however,

is the opposite - the background looks stable and the target appears to move.

What then is the coordinate system for the intermediate-level system? A purely head-based system leads to difficulties during pursuit eye movements, when the object of interest is moving in head coordinates. Consider a scene consisting of a side view of a person riding a bicycle. In the frame of the bicycle the cyclist's motions are easy to describe; her feet are moving in circles while her knees oscillate along an arc centered on the hip joint. In a head frame, these motions become complicated cycloids. This presents serious problems for a connectionist model. In order to keep the size of the networks reasonable it is essential that the set of primitive trajectories be describable by a small number of parameters.

In the Four Frames model the pursuit problem is dealt with by postulating a special 'pursuit mode' during which indexing proceeds straight from the retinal frame to the World Knowledge Formulary (*i.e.* the relational model level) without passing through the head frame. This solution assumes that all the appropriate indexing information is available in the retinotopic frame. The assumption is reasonable for segmentation information, since the retinotopic frame has all of the information needed to segment the scene near the fovea, and in fact will have higher resolution near the object being tracked. For trajectories, however, it presents problems. Computing trajectories requires integrating information across large spatial and (more important) temporal intervals. We have assumed that the low-level systems do not preserve information for more than 100 milliseconds, so trajectory computation at the low level would be very difficult.

As an alternative to a separate pursuit mode, suppose that the gaze mapping that connects retinal to stable coordinates is programmable, rather than being strictly controlled by eye position[3]. In static vision, the gaze mapping would default to the standard retina-to-headframe transformation. During pursuit it would track the object being pursued. In the case of the hypothetical cyclist discussed above, this would allow an observer, to make saccades between the front and back wheels without having to resegment the scene or reanalyze the trajectories of the legs. The notion of a head frame is replaced with an attentional coordinate system, in which the observer constructs whatever retinal-to-stable coordinate mapping is appropriate for the scene and perceptual task.

Facilitation Over Time

It is often noted that apparent motion stimuli produce a stronger motion percept when they are presented repeatedly – for example, by presenting the two frames that make up a stimulus in alternation [Kolers, 1972]. In the model this effect arises because of top-down priming from the scenario level. When a repetitive stimulus is presented, the high level system constructs a description of the sequence of events

[3]I am indebted to Mary Hayhoe for this suggestion.

over time. Following Goddard, one can think of this as a finite state machine. This machine allows the system to predict what should happen next - where the dots now in view will go when they disappear. This prediction will prime the intermediate level, causing it to relax more quickly to a stronger solution.

4.3 Modelling the Architecture

We have seen that the architecture sketched in section 4.2 is computationally reasonable and consistent with the biological facts, to the extent that it is detailed enough to evaluate on either ground. We will now examine how well existing models of motion processing fit the architecture and see where work remains to be done. At each level of the architecture two questions must be asked. First, are there existing models that compute roughly what the architecture says they should? Second, can those models be implemented using biologically plausible mechanisms?

The Low Level

A look back at Chapter Three shows that there are many models compatible with the low level of the architecture - that is, that compute approximations to the optical flow field by analyzing contrast changes over short time intervals. In section 4.2 it was argued that very accurate computation of optical flow was impractical and unnecessary. For this reason we favor simple models such as the oriented spatiotemporal receptive fields of [Adelson and Bergen, 1986] or the slightly more elaborate local flow estimators of Heeger [Heeger, 1986]. The claim is that these algorithms are good enough for our purposes - that although they do not give very accurate flow maps, they do as well as is necessary to support the intermediate and high levels.

Unfortunately the situation is not as good when it comes to segmentation. The model requires some process that maps the retinotopic features of the low level to the non-retinotopic shape primitives of the intermediate level. At present there are no algorithms that can do this at all robustly. Segmentation is undergoing a resurgence in interest, however, and some promising work is being done. Pentland's attempts to recover superquadric primitives from very limited information are meeting with some success [Pentland, 1988], and various researchers are developing principled ways to combine information from maps of different low-level features [Chou, 1988].

The High Level

A number of computer vision researchers have worked on ways of recognizing figures from Johansson-style moving light displays [Rashid, 1980; Hoffman and Flinchbaugh, 1982]. Their approach was purely abstract, however, making no concessions to the underlying hardware. Goddard's model, on the other hand, is an excellent fit to the

conceptual architecture. This isn't surprising, since the high level of the architecture is based on his ideas. In any case, his model is computationally sound, is implemented in a connectionist network, and assumes exactly the intermediate-level representation called for by the architecture. It is also capable of supplying top-down reinforcement to help the intermediate level converge to a useful solution.

The Intermediate Level

It is at the intermediate level that existing models are least satisfactory. Long-range motion is the primary probe of the intermediate level, and existing models of long-range apparent motion do not account very well for the phenomena as described in Chapter Two. The defects include

- **incorrect primitives**
  Many models are based on very low-level primitives, such as Marr's primal sketch tokens. According to the architecture the primitives should be at the level of segments.

- **incomplete representations**
  The intermediate level is supposed to compute trajectories, represented as motion at constant speed along a path. Most models have no explicit representation of trajectory.

- **failure to deal reasonably with time**
  A number of models ignore time altogether, and hence cannot explain such first-order effects as Korte's third law. Even those that do handle time often divide the image sequence into frames, and consider their job done when they have computed correspondences between the old frame and the new one. The division into frames seems highly unnatural, and begs the question of how the division is done when the world is changing continuously.

Not all of the models have all of the defects, but most have more than one.

Many of the weaknesses of existing apparent motion models can be traced to a failure to think of apparent motion as an aspect of normal perception. The domain of typical apparent motion stimuli is highly constrained, so it is not surprising that models designed principally to handle that domain make little sense in a more general vision context. If one only worries about processing tachistoscopic presentations, there is no reason not to divide the world into frames. If one does not think about what apparent motion is *for*, it is easy to ignore the evidence that trajectories are part of apparent motion percepts.

Not all models suffer from this philosophic difficulty. In particular, the optic flow algorithm of Yuille and Grzywacz [1988] incorporates Ullman's minimal mapping theory into the normal process of optic flow recovery. Other difficulties of the minimal

mapping theory remain, of course. The Yuille and Grzywacz story, interestingly enough, reverses the hierarchy of computations called for here: they use long-range motion to assist optical flow recovery, while we claim that optical flow estimates are primarily useful as a means of helping us construct the intermediate-level motion representation.

Conclusion

We have seen that reasonable models exist for the high and low levels of the proposed architecture. The problems at the intermediate level are twofold. First, no one knows what sort of segments (parts, shape primitives) the intermediate level uses, or how it extracts them from the low-level feature maps. Second, there is no good model of how the intermediate level assigns trajectories to the primitives. The first problem is (in the author's opinion) *the* fundamental problem of precategorical vision, and has proven to be extraordinarily difficult. Any principled attack on it would require broadening the scope of the work to include a theory of static recognition, detailed models of the feature maps, and much more that is far beyond the scope of a theory of motion perception. The rest of this thesis, therefor, will be an attempt to make progress on the second front. We will try to come up with a model of intermediate-level motion perception that avoids the problems of the apparent motion models discussed above. In the next chapter we will approach the problem rather abstractly, focussing on the kinds of things that must be represented, the constraints imposed by connectionist modelling rules, and the alternatives for representing and computing the necessary information. These ideas will then be used to develop a model that is detailed enough to be built, and that can be demonstrated on real apparent motion stimuli.

5 Toward a Model of Intermediate-Level Motion Perception

In Chapter Four we argued for a three-level architecture for the human motion understanding system. The most novel aspect of that architecture is the intermediate level, at which trajectories are assigned to shape primitives extracted from low-level feature maps. This chapter will examine the computational problems involved in constructing the intermediate level representation. As stated at the end of Chapter Four, we will avoid the question of how the scene is segmented into shape primitives, on the grounds that it is too hard and strays too far from the central topic of motion understanding. Instead the focus will be on finding ways to assign trajectories to the segments after they have been extracted. The price of this simplification is that we will unable to say much about interactions between the trajectory-assignment process and segmentation or grouping processes. Such displays as Ternus' stimulus (see Chapter Two) suggest that interesting interactions do take place. However, there seems to be no way to avoid making the simplification without going far beyond the scope of the present problem.

In this chapter we will continue to talk in general terms, avoiding premature specification of the details. The goal will be to present a general strategy for dealing with intermediate level motion and then examine the major design decisions that will have to be made in order to build a working system.

The next section discusses the particular connectionist formalism that will be assumed for the rest of the thesis. We will then review the properties and behaviors that are characteristic of intermediate level motion, drawing on the architecture of Chapter Four and the literature review in Chapter Two. The last (and longest) section will present an overview of the model followed by a discussion of the connectionist representation problems that must be solved in order to make it work. In some cases we will present several alternative solutions, selecting one to be developed in detail in Chapters Six and Seven.

5.1 The Modelling Formalism

A philosophical stance taken in Chapter One was that the ultimate goal of this work is a model that could (at least in principle) be implemented in the brain, *i.e.* by neurons. Our method of reaching this goal is to restrict consideration to models expressed in a connectionist formalism. Later sections of this chapter will consider detailed mechanisms for representing various types of information. Before doing that it is necessary to commit to a particular set of connectionist design rules, so that the hardware constraints on the design will be clear. We will also present some of the philosophy behind the choice of formalism and explain what it is that we hope to get out of it.

5.1.1 Why a Connectionist Model?

The basic reason for developing the model in connectionist terms is to gain a degree of biological plausibility. Using a formalism that shares many of the computational properties of neurons will make it more credible that the prescribed computations could actually be performed by neurons. A second motivation is that connectionist models are interesting as abstract models of computation. Working with them may yield interesting insights about parallel computation.

Connectionism and Biological Plausibility

Modern connectionism has its roots in the observation that brain functions can be usefully described as computation. In the particular case of perception, one attempts to infer facts about the state of the world from the effects of that state on the sensorium. However, the brain is not very much like a conventional computer. Computers are made of transistors that can switch very quickly, but are designed so that only a tiny fraction of those transistors switch at each time step. Brains are made of neurons that change state much more slowly, but their computation proceeds much more in parallel. Computers can copy elaborate symbolic structures by moving a few pointers around; neurons can transmit only a few bits of information per millisecond, and their connections are fixed.

Of course one can argue – and people do – that since both brains and conventional computers are Turing-equivalent, it doesn't matter what formalism one uses. In this view connectionist models are an implementation detail, with no influence on the computational and algorithmic aspects of the problem. This argument founders on two observations. First is the 'hundred-step rule' [Feldman and Ballard, 1982]. It takes on the order of 5 milliseconds for a neuron to transmit a useful amount of information (*i.e.* a few bits). Humans can perform complex perceptual tasks such as recognition in about 500 milliseconds. Therefor, the computation cannot possibly

require more than 100 communication cycles from start to finish. This is a very tight constraint on the classes of models that are plausible. If one is serious about wanting to know what algorithms the brain uses, it is essential to use a computational formalism that makes the number of communication cycles explicit.

The second objection to the 'mere implementation' argument is that it ignores the empirical fact that one's choice of programming language has an enormous impact on how one thinks about computational problems, and on the types of solutions one arrives at. By training ourselves to describe computations and algorithms in terms of massively parallel networks, we hope to gain new insights into the nature of the perception problem and to be led to new approaches to it.

5.1.2 Choosing a Formalism

Connectionist models now come in an assortment of flavors and styles. There has been an unfortunate recent trend toward identifying one style or another as the One True Faith; in fact each has its strength and its natural domain. The major subdivisions are:

- Detailed neural models, using the best available information about real neurons work. The goal here is to model neural, dendritic or synaptic circuits in detail, to get at the specifics of such phenomena as direction selectivity, memory et cetera. Examples include the work of [Torre and Poggio, 1978].

- Unstructured or PDP models [Rumelhart and McClelland, 1986b]. It is this version which has received the most attention during the recent wave of interest in in connectionist models. The focus here is on learning and pattern recognition, and on the construction of distributed representations. The domains chosen are usually either entirely abstract (*i.e.*, directed at the question of what a particular model is capable of learning or representing) or taken from areas in which the neural representations are completely unknown, such as language and cognition.

- Structured models [Feldman and Ballard, 1982]. The emphasis here is on design rather than learning, connectionist principles being used to constrain the set of possible designs. A fair amount of work in this area relates to vision, and it may be that the extraordinarily rich structure of visual cortex influenced the development of the paradigm. Other researchers have used it to tackle problems where unstructured models seemed unrealistically simple, such as property inheritance and inference in a knowledge hierarchy [Shastri, 1985].

The work to be described here is based on the structured formalism of [Feldman and Ballard, 1982]. Clearly, the state of current knowledge would not permit development of a detailed neural model. A PDP-style model would be interesting,

but would tend to shift attention away from the central issues of what to represent and how to compute it. In particular the PDP emphasis on learning representations would make it difficult to explore the design space and see what sorts of representations are possible. Finally, working in a structured formalism will make it possible to build on related vision work in the same paradigm [Feldman, 1985; Ballard, 1984].

It is important to be clear about what we expect to gain by using this formalism. The computational units that will be used to build the model are not intended to represent actual neurons, but rather to capture their computational properties in an abstract form. It is assumed that the need for neurons to communicate with each other dominates the character of neural computation. If this is so, then it doesn't matter (within limits) what individual units are allowed to compute, as long as their communications are restricted in the right way. (For a more detailed justification of the formalism see the Feldman and Ballard paper.) The experience of structured connectionists to date has been that this formalism does indeed place strong constraints on what can be represented and computed. The constraints in general come from the following facts:

- Connections are fixed; they cannot be change while the network is running.
- Networks must be highly parallel in order to satisfy the hundred-step rule mentioned above.

These two facts force the use of representations and algorithms that require large numbers of units. In consequence it becomes important that

- The brain has a finite number of neurons – on the order of 10^{11}.

In other words, resource constraints tend to drive the design.

The networks developed below will consist of simple *units* that communicate over one-way *connections*. Each unit will maintain three pieces of information: its *state*, its *potential* and its *output*. The state is an operating mode defining how the unit responds to its inputs, such as *e.g.* NORMAL or FATIGUED. The set of states for a unit is assumed to be small – it is not to be used as a memory for the unit. The unit's potential is a real number on some bounded interval. The output is an integer whose range is limited to reflect the limited bandwidth of neurons. In the original formalism outputs were taken from the range $[0, 10]$, but we will relax this here to allow up to ten bits of precision.

In addition to its state information each unit specifies a set of functions that define how the next state, potential and output are computed from the current state, potential and inputs. We place no formal restriction on the unit functions, requiring only that they should satisfy some intuitive notion of simplicity. The intent is that a

unit should not be able to compute anything that could not plausibly be computed by a small number of neurons. We do allow a number of things forbidden to PDP models, though. Most important is that units are allowed to treat each input differently – they are not restricted to simple functions such as weighted sums and products. For example, units will be allowed to compute boolean functions of their inputs, or use one input to modulate their response to others. In order to make this easy to express we identify a set of *input sites* on each unit, and associate each input with a site. Each site computes a simple function of its inputs using a *site function*, and it is these site values that are made available to the potential, state and output functions.

A network is completely defined by the set of units and sites, the set of updating functions for each unit, and the pattern of connections between unit outputs and site inputs. Computations proceed as follows: At the beginning of each time step, every unit broadcasts its output value to every input that it is connected to. Next, every site looks at its inputs and computes a value using its site function. Finally, every unit computes new state, potential and output values based on the current values plus the values of the sites.

5.1.3 Connectionist Methods

Structured connectionist techniques will be discussed in more detail in the context of particular representation problems later in this chapter. It will however simplify that discussion if we take a moment to review a few basic terms and techniques.

Value Unit Representations A continuous parameter is said to be represented using *value units* if one unit is dedicated to representing each possible value of the parameter. See [Ballard, 1984] for a discussion of the virtues of such a representation.

Hough Transforms The Hough transform is a standard technique in computer vision, used to extract parameters of some high-level phenomenon of interest from observed features that constrain those parameters. In the classic application, we observe local edges, each of which is compatible with a small set of line parameters. Each edge 'votes' for consistent line parameters, and lines that receive many votes are accepted as being present in the image.

Hough transforms map in a straightforward way to connectionist networks. Given value unit representation of the features and the output parameter space, we simply make a link from each feature to all consistent output units. The output units compute a sum, and fire if their sum is over some threshold. This description glosses over many subtleties, some of which we shall say more about below. Let it suffice to say that the technique is fundamental to a great deal of connectionist work, particularly in the area of vision. Ballard in particular has elaborated the idea into large-scale theories of

perception and cognition, and has done much to extend and generalize the technique in various directions [Ballard, 1981; Ballard, 1984; Ballard, 1986a].

Conjunctive Connections A conjunctive connection is essentially a logical AND of its inputs. It is often used in conjunction with the Hough transform, so that a given feature casts a vote only if some other consistent feature is present [Ballard, 1984]. It can also be used to construct temporary associations between subnetworks, as in the two-out-of-three binder units of [Feldman, 1982].

5.2 What the Model Must Do

We will begin our discussion of modelling intermediate-level motion by presenting a loose specification for the model. The idea is to draw together material from Chapters Two and Four in order to constrain the set of solutions as tightly as possible.

5.2.1 Inputs

The basic input to the intermediate level will be a representation of the scene in terms of intermediate level primitives, *i.e.* some type of segments. The representation is image-like in that it is indexed by x and y, but it is in an unspecified non-retinotopic coordinate system, and the objects represented are complex descriptions of the primitives. It can be thought of as an array of property vectors, each describing the segment centered at a particular location. The rather generous assumption will be made that the segmentation system abstracts segments that may have arbitrary spatial extent in the low-level representation to points in the intermediate-level representation.

Like the segment representation, the choice of a segment properties will necessarily be arbitrary. At the least the description will be assumed to contain an encoding of any retinal slip information associated with the segment, as well as gross properties such as rough shape, depth, color et cetera.

As time proceeds segments will appear and disappear at various locations in the array. It is important to realize that segments are no more than lists of properties associated with particular locations. They do not have identities, and there is no explicit representation of such ideas as "the segment appearing at point B corresponds to the same object as the segment that recently disappeared at point A." Inferring and representing such facts is the job of the motion system. Intermediate-level motion processing can thus be viewed as a sort of temporal analog of segmentation. Where spatial segmentation asserts a common identity for surface patches at different locations, intermediate-level motion analysis asserts the identity of segments across time.

5.2.2 Outputs

The intermediate-level motion system's job is to associate trajectories with each segment in the scene. Trajectories will be represented in the intermediate-level coordinate system, which (as per Chapter Four) can be head-based, object-centered or retinotopic as the observer's goals dictate. The nature of computing with connectionist nets requires that the trajectory representation be chosen from some primitive set. We shall assume that the representation is in terms of constant rates of speed along some set of paths.

What is the set of paths? Apparent motion data suggest that they include motion in depth [Attneave and Block, 1973] along simple curves [Shepard and Zare, 1982]. The set of circular arcs in 3-space is important for recognition of moving light displays, since animal limbs can be approximated by rigid rods joined at their ends. Also, if a point on a rigid object is fixated, then the motions of its parts can be approximated as arcs as long as it is not moving rapidly in depth [Bandopadhay, 1986]. Thus the set of circular arcs in three-dimensional space would be a reasonable minimal set of primitive trajectories Of course, it would not be surprising if trajectory primitives could be learned. That would explain how people can become so good at predicting the positions of tennis balls, juggler's clubs et cetera.

5.2.3 Behavior

The most fundamental requirement for the intermediate-level motion system is that it interpret simple continuous motions correctly. That is, given a segment moving smoothly across the input field, generating appropriate retinal slip as it goes, the system should easily infer the correct trajectory and speed. It should do about as well (but perhaps with a longer integration time) when the stimulus generates no retinal slip, as in the case of disparity, "non-fourier", or isoluminant stimuli. This implies, if it wasn't obvious before, that the system cannot depend solely on direction selectivity inherited from the low level. Its units must directly detect the spatiotemporal arrangement of stimuli.

The most interesting requirements come from the claim that intermediate-level motion processing continues to work under the extremely impoverished stimulus conditions of apparent motion displays. The apparent motion literature was reviewed in Chapter Two; let us summarize the major characteristics and phenomena with a view toward identifying properties of the intermediate-level motion system.

Gestalt character

First and foremost, apparent motion is a Gestalt phenomenon. It imposes order on scenes which initially seem to be disorganized. Interpretations tend to be discrete;

given a stimulus in which correspondence is ambiguous, for example, observers do *not* see a linear combination of the possible interpretations. Instead they see one interpretation and all others are suppressed. This intelligent-seeming property of the system made apparent motion displays popular with the perceptual inference school in psychology [Rock, 1983; Gregory, 1970].

Of course, the discrete nature of apparent motion percepts does not imply that the underlying computation is an inference process – otherwise one could conclude that processes like stereo fusion were also high-level computations. It does indicate that there is competition between rivalrous interpretations. The claimed 'cleverness' of the interpretations indicates that the system has subtle and sophisticated measures of how good a given interpretation is, and of how consistent two partial interpretations are. Any model of intermediate level motion must provide mechanisms for interpretations to compete, and must allow a wide variety of information sources to influence the competition.

Importance of time

For any given spatial separation, there is a limited range of temporal asynchronies that will give rise to an apparent motion percept. The lower limit of the range varies linearly with spatial separation measured along the perceived path [Shepard and Zare, 1982; Attneave and Block, 1973; Farrell, 1983]. As many have observed, this has a natural interpretation if one assumes that motion is represented in terms of constant speed along a path, as in the proposed architecture. It also confirms the claim made above, that the system must have available to it a representation of the asynchrony between two stimuli as well as the spatial separation.

Interpretation Cues

A wide variety of cues influence the interpretations given to ambiguous displays. In most apparent motion situations there are two distinct types of ambiguity present. First, there is the question of how the stimuli in one frame correspond with the stimuli in the next. Second, there is the question of what trajectory a stimulus traversed to get from one location to another. Factors that are used to resolve the ambiguity include

- **attention and bias**
  Both types of ambiguities are strongly influenced by expectation and attention.
- **spatial separation**
  Correspondence ambiguities are dominated by preference for nearest neighbors.
- **retinal slip**
  Information from the low-level system affects both types of ambiguity.

- **shape and other properties**
  It is generally agreed that shape, color et cetera have weak but noticeable effects on correspondence ambiguities. That is, stimuli tend to be matched in ways that allow the properties of the (hypothesized) underlying objects to remain stable over time. Foster's stimulus demonstrates compellingly that shape has a strong impact on trajectory ambiguity.

Any serious apparent model must provide ways of taking these influences into account.

5.3 Overview of the Model

We will begin the discussion of a model of intermediate-level motion by looking at the general character of the model. Figure 5.1 shows an idealized view of what the model should look like. The model is divided into two groups of units, one representing the input segment descriptors and one the output trajectories.

Input

The input segment descriptors will be represented by vectors of properties at each possible segment location. Note that this is not a value-unit representation, since the model does not have a single unit for each possible combination of properties and locations. Instead, several units must be turned on to represent a given segment. This representation is necessary to avoid combinatorial explosion, as argued in [Feldman, 1985].

The set of possible segment locations obviously includes all (x, y) locations in the intermediate level frame, at some level of quantization. An important question is whether segment locations should be indexed by depth as well. At first it might seem that adding another spatial dimension to the representation would make the network impractically large, but this is not necessarily so. The quantization in depth could be quite coarse. If the intermediate-level system uses an 'attentional' coordinate frame, as suggested in Chapter Four, then the quantization could be very coarse indeed. Quantization into three values would give depth measures such as 'canonical', 'nearer', and 'farther'. A few more depth values would be enough to support simple inferences about occlusion or the likelihood of collision between two moving segments.

Output

The output trajectory set clearly includes straight lines and simple curves at constant arc speeds, and apparently does not include arbitrary paths or accelerations along paths. For the moment we will assume a value-unit representation for the trajectories. There will be a unit for every possible combination of starting location, ending location, connecting path and speed.

The Computation

The basic computation performed by the network will be a type of Hough transform. That is, each segment descriptor will send excitatory signals to units representing trajectories that it is consistent with. In some sense this is easy – a given segment is consistent with all trajectories passing through its location. A network wired this way would in fact be able to pick out arrangements of segments into line and arc formations. It would not, however, be able to capture any of the temporal properties of motion perception, simply because the network does not yet have a representation for time.

5.4 What's wrong with this picture?

There are three basic problems with the simple network sketched above. First, there is the problem of representing the output parameter space, *i.e.* the trajectories. Combinatorics may not permit a true value unit representation. Second is the time problem. We must find ways for the Hough transform voting process to take into account the spatial and temporal relations between stimuli. Finally, there is the question of how to handle the discrete 'Gestalt' quality of apparent motion percepts. Hough transform voting produces a pattern of activation over the output parameter space, which is a poor model of what goes on in intermediate-level motion analysis.

In the remainder of this chapter we shall look at ways of dealing with these problems.

5.4.1 Representing Trajectory Space

The first problem that must be dealt with is how to handle representation of trajectories. The central question here is whether it is reasonable to propose a value unit representation of the space, *i.e.* a unit for every possible trajectory. For concreteness, consider a minimal set of trajectories: the set of all straight and circular arc segments. Suppose the segments are indexed by starting point, ending point, curvature and speed. If starting and ending points are arbitrary locations in an $n \times n$ array and there are C curvatures and S speeds, the parameter space will require CSn^4 units. Suppose $C = S = 10$. Then for $n = 10$ we get 10^6 units, which is large but not at all impossible. For $n = 100$ we would need 10^{10} units, which is perilously close to the total number of neurons in the brain. For reference, Goddard's current motion-based recognition network assumes a 30.5 by 24 input array, requiring 5.3×10^7 units. If the trajectory set is extended to include depth and/or a richer class of curves, it seems almost certain that the value unit representation will require too many units.

Counting problems like the above are frequently encountered in connectionist models. Consequently, many researchers have spent time developing general techniques

for dealing with them. All techniques amount to methods of encoding or distributing the representation. so that individual units represent overlapping regions of parameter space rather than points in it. The result is that each point in the space is represented by a conjunction of units in the network. The price paid for the saving in number of units is that representations of multiple points in the parameter space may interfere with each other. Among the standard techniques are:

- **Coarse-Fine Coding**
  Figure 5.2 a) illustrates the principle of coarse-fine coding. A single n-dimensional array is replaced with n arrays that are finely quantized in one dimension and coarsely quantized in the others. Conjoining one unit from each array, chosen so that all of their coarse dimensions overlap, specifies a single point at full resolution.

- **Interpolation Coding**
  Another approach is to let individual units stand for widely separated points in the parameter space, and represent points between them by partial activation of units at these 'grid points'. This approach has an obvious analogy with the human visual system's representation of an enormous number of colors by three classes of cones. Ballard [1986c] analyses the general case and shows that even restricting unit outputs to ten discriminable levels, this method can represent points in the space very accurately at relatively low cost.

- **Projections**
  A limiting case of coarse-fine coding is to let the units have no specificity at all along one or more dimensions. That is, the network only represents projections of the parameter space into lower dimensional subspaces. If necessary a coarse representation of the whole space can be used to help tie the projections together. Ballard [Ballard and Sabbah, 1983; Ballard, 1984] presents a number of demonstrations of the technique. Figure 5.2 b) provides an illustration.

5.4.2 Representing Time

A second and deeper problem for the model sketched above is that of making the network sensitive to the spatial and temporal relations between stimuli. As described so far the network would be unable to distinguish between a segment moving smoothly through a sequence of locations and one jumping randomly between members of the same set. The basic computational paradigm is the Hough transform, in which features 'vote' for higher level structures. What must be done is to alter the design so that a segment only votes for a trajectory if there are other segments around whose relative positions in space and time are consistent with that trajectory. Another way to state this is to say that the Hough transform features are not the segments themselves, but pairs of segments in appropriate spatiotemporal relationships. Thus

the network will make use of a complex type of conjunctive connection, in which each conjunct requires not only that both features be present, but that their relative ages be consistent.

In order to continue, then, we need a way to build conjunctive connections that are sensitive to the relative ages of their inputs. That in turn requires development of a way to represent the ages of segments. In apparent motion it appears that the age of a stimulus is measured from its onset – as stated in Chapter Three, stimulus onset asynchrony is the critical variable in Korte's Third Law.

Time in Connectionist Nets

Unlike the combinatorics problem, the problem of representing and using temporal information has received little attention from the connectionist modelling community. Even modellers of inherently sequential phenomena such as language have frequently assumed that the entire input is presented statically. By doing so they convert temporal locations to spatial ones, dedicating a separate group of input units to represent each succeeding moment in time.

This is not to say that the temporal behavior of networks has been entirely ignored. In the realm of unstructured networks using highly distributed representations, a number of people have built networks in which some 'hidden layer' units are not connected to the input or output, but only to each other [Jordan, 1986; Elman, 1988]. The result is that the network can save internal state information for future use. Such networks, for example, can learn to recognize and reproduce sequences of inputs.

Encoding Stimulus Age

Time is a dimension like any other, so the methods used to represent it are similar to connectionist methods used for other parameters. It must be treated carefully, though, since the networks themselves exist in time. Among other things one must worry about how continuously changing inputs will affect the convergence behavior of the networks.

Discrete Representations The simplest approach to representing stimulus age would be a straightforward application of the value unit principle. One might construct a network like that of Figure 5.3, which will be called a ripple clock. A ripple clock is simply a chain of units with particular activation rules. The first unit in the chain is activated by some triggering event. It becomes active for some short interval and then stops. The next unit in the chain treats the previous unit's cessation of activity as a trigger and begins firing in turn. Thus the initial trigger causes a wave

of activity to ripple down the chain at a rate determined by the firing interval of the units.

How many units are required to build a ripple clock of reasonable size? Assume that the time representation needs to be accurate to within 10%, which is close to McKee's measured value of 8% at 480 msec [McKee, 1981]. Each unit in the ripple clock can be thought of as representing a temporal "window" whose width is about 10% of its canonical value. Covering a range of one order of magnitude, therefore, would require k units such that $1.1^k = 10$. Solving for k shows that about 25 windows of width 10% are needed to span a factor of 10. For example, one order of magnitude would span the range from 50 to 500 milliseconds, which is comparable to the range of times over which apparent motion occurs.

Although ripple clocks work, they require a rather large number of units. Furthermore, they are somewhat expensive to use for our purpose. Recall that the motion network needs connections which prefer a particular temporal *difference* between each pair of clocks. To do this with ripple clocks one would need a conjunctive connection for each possible clock value. For example, a unit that expects to see a difference of two time units between locations A and B might compute

$$(t_{A1} \wedge t_{B3}) \vee (t_{A2} \wedge t_{B4}) \vee \ldots$$

The same unit will presumably need to look at many pairs of locations, so the number of connections may become a serious problem. Another objection to ripple clocks is that they do not seem at all biological. No neurons with such discrete behavior have been reported in cortex. The same objection applies to encodings of ripple clocks. One could obviously build a conventional binary counter out of connection units, but would be very surprised to find such a representation used in the brain.

Level Encodings The alternative to value unit approaches (and encodings thereof) is to use clocks whose transient responses encode temporal information. That is, after triggering by stimulus onset, this type of clock unit's activity is some predictable function of time. The simplest version of this is to raise the unit's output to a fixed level when it is triggered and let it decay monotonically. For example, suppose that the output is initially set to one and decays at a constant rate r. The output of the unit is then $1 - r(t - t_0), t < t_0 + 1/r$, where t_0 is the time of stimulus onset.

We need a way for input sites on a trajectory unit to respond in proportion to how well the time difference between two stimuli agrees with the expected value for that trajectory. (The expected value is known because the trajectory encodes path curvature and speed and the particular input sites encode location. The expected time difference is simply arc length divided by speed.) Suppose that stimuli appear at time t_1 at location 1 and time t_2 at location 2. The clock units C_1 and C_2 will produce linearly decaying outputs as specified above. Let the site function for a

connection with expected time difference t_e be

$$\begin{aligned} G(C_2 - (C_1 + t_e)) &= G((1 - r(t - t_2)) - ((1 - r(t - t_1)) + t_e)) \\ &= G(r(t_2 - t_1) - t_e) \end{aligned}$$

where G(t) is some function that distributes activity around 0, such as a Gaussian or triangular pulse. Varying the width of G determines how strongly the site responds to time differences that do not exactly meet expectation. Figure 5.4 illustrates the behavior of the decay clock.

Simple variants of the decay clock would obviously work just as well. For example, one could replace linear decay with exponential decay and turn the site function's differencing operation into a ratio. Decay clocks seem quite reasonable from a biological point of view. It must be admitted however that although decaying response after stimulation is a property of many cortical neurons, it is not predictable enough to make a particularly good clock. At the least one would need to average over substantial populations to get reasonably accurate results. Also, accounting for apparent motion data would require clocks that continue to respond for up to one second, which is longer than the decay typically observed.

Clocks based on temporal response functions can be elaborated almost arbitrarily. Suppose, for example, that we use pairs of clocks at each location emitting damped sine and cosine waves:

$$\begin{aligned} C_X &= d(t - t_x)\cos(t - t_x) \\ S_X &= d(t - t_x)\sin(t - t_x) \end{aligned}$$

Taking products of some of the terms and subtracting gives:

$$\begin{aligned} S_1C_2 - S_2C_1 &= d(t - t_1)\sin(t - t_1)d(t - t_2)\cos(t - t_2) \\ &\quad -d(t - t_2)\sin(t - t_2)d(t - t_1)\cos(t - t_1) \\ &= d(t - t_1)d(t - t_2)(\sin(t - t_1)\cos(-t + t_2) + \sin(-t + t_2)\cos(t - t_1)) \\ &= d(t - t_1)d(t - t_2)\sin(t_2 - t_1) \end{aligned}$$

One can recover the cosine similarly, divide to eliminate the decay function $d(t)$ and estimate the time difference from the inverse tangent. The technique can be varied in a number of ways, for example by producing what amounts to a discrete version of the spatiotemporal energy models of [Adelson and Bergen, 1986].

In conclusion we can say that there are a number of ways to represent the relative ages of stimuli using predictable unit temporal responses. They can be used to build site functions sensitive to relative ages of stimuli, using only weighted sum and product operations plus simple non-linearities.

5.4.3 Gestalt Behavior

The last major problem that must be solved in order to turn our architectural sketch into a workable system is that of capturing the Gestalt characteristics of apparent

motion percepts, their discreteness and their tendency to impose order on scenes. In many ways this is the most interesting computational aspect of our problem. Similar grouping and organizing phenomena are fundamental aspects of perception, both in vision and in other senses. Any advances in this area will therefor have wide applicability.

The behavior that we are trying to capture is a little hard to define compactly. There are a number of ways to describe it, all of which overlap a bit but fail to entirely span the space. Perhaps the simplest characterization is to say that *inconsistent explanations inhibit each other.* The system's interpretations are usually coherent in the sense that they do not imply mutually exclusive assertions about what happened. Another way to express this is to say that *interpretations are unambiguous.* Each stimulus is assigned a single role, and there is no need for any subsequent thresholding or other analysis of the output. Interpretations are discrete, having the flavor of logical assertions about what happened, rather than (possibly conflicting) evidence about it.

The Hough transform approach described so far clearly does not have the desired character. In the most straightforward version of the algorithm 'interpretations' (*i.e.* output units) do not compete, nor is there any notion of consistency between them. Also, Hough transforms are ambiguous in that the end product of the voting step is a distribution of votes over the output parameter space. An additional step is required to interpret the vote distribution in terms of assertions about what is or is not present in the input. What is needed, clearly, is a way to bring some of the character of constraint satisfaction algorithms to the Hough transform. We want to preserve the simple structure of Hough transform networks, while adding the ability for interpretations to detect inconsistencies with each other and to compete based on the available evidence.

The next chapter presents a mechanism called *feature binding* that has the desired properties. The mechanism itself is simple; however, analyzing it and comparing it with alternative approaches requires a lengthy discussion. Also, the argument is completely independent of the motion problem and is hence somewhat removed from the main thrust of the thesis. For these reasons it is discussed in a separate chapter.

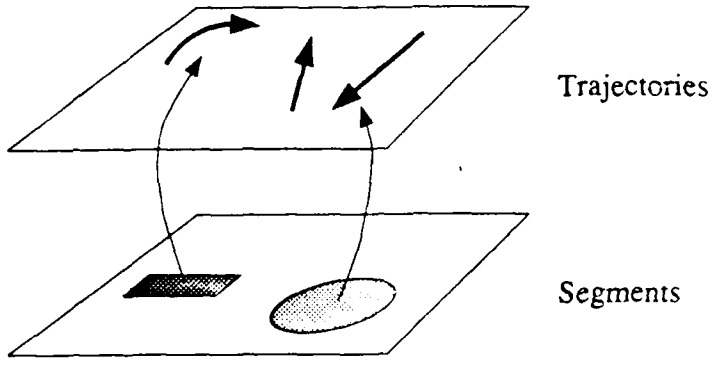

Figure 5.1: Structure of the intermediate-level motion model.

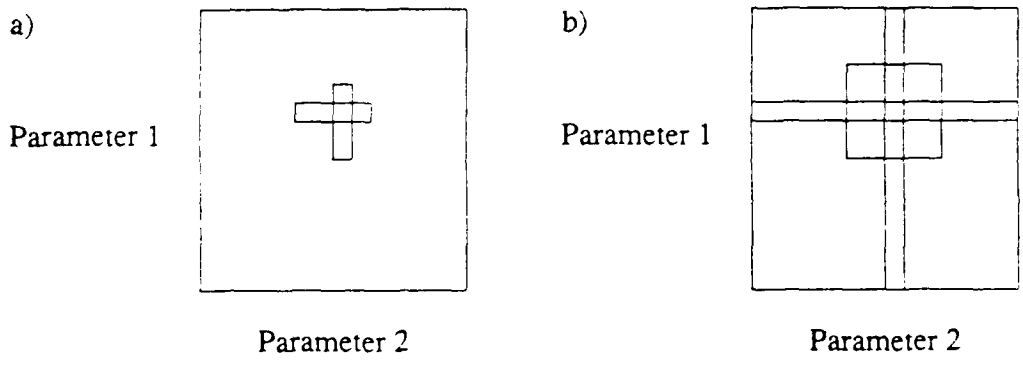

Figure 5.2: Parameter space encodings. a) Coarse coding: a k-dimensional point is represented by the conjunction of k units, each finely tuned in one dimension and coarsely tuned in all others. b) Loosely coupled subspaces: a point is represented by the conjunction of finely tuned projections of the space with a coarse representation of the full space.

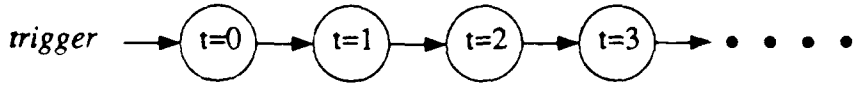

Figure 5.3: Ripple Clock

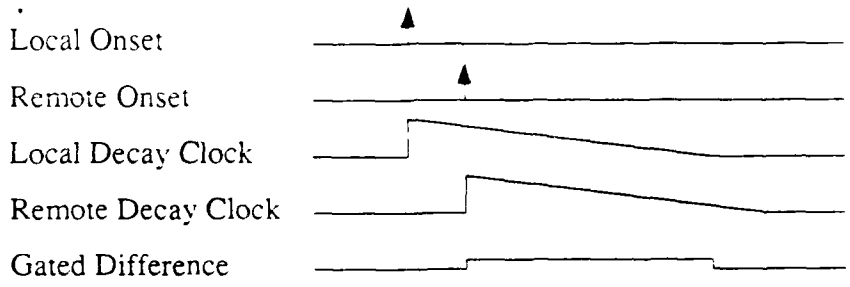

Figure 5.4: Response of a decay clock, and difference of two decay clocks triggered at different times.

6 Feature Binding

One of the most striking properties of apparent motion percepts is their gestalt character. At the end of Chapter Five we left open the question of how to capture this behavior in the intermediate-level motion network. The problem is both more complex and more interesting than the other representation problems discussed there. In part this is due to its generality – it is a problem that arises over and over in the study of perception. In this chapter, therefor, we will temporarily step back from the specifics of motion understanding and look at the problem in the abstract. The goal will be a general method of obtaining gestalt-like behavior in simple Hough transform networks. Any modifications needed to apply the technique to the motion problem will be made in Chapter Seven, when we present a simple but completely specified motion understanding network.

In the next section we will discuss the type of behavior that is needed and argue for a particular way of defining the problem. We will also review related connectionist and computer vision work. Section 6.2 will present the general feature binding strategy, an algorithm that implements it, and an analysis of what the algorithm is really computing. The last section will deal with the application of the algorithm to general Hough transform applications and discuss variants and extensions to the method.

6.1 Problem Definition – What is Needed

The formation of a visual gestalt is one of the more impressive phenomena in human vision. A collection of low-level visual features seems to spontaneously organize itself into a coherent whole that was not inherent in the parts. The parts may gain properties that weren't visible in isolation – for example, they may suddenly appear to be at different depths. The interpretations assigned to features can be strikingly complex. In apparent motion, for example, objects can move behind one another, or deform, or flip out of the image plane. In some cases the same visual stimulus can be organized in more than one way, as in the famous Necker cube, and in these cases the properties of individual features appear to change when the observer switches from

one interpretation to another. Despite the ambiguity of the stimulus, observers can generally only perceive one interpretation at a time. Many gestalt phenomena share these properties: they assign roles to primitives, enforce global consistency, and result in single, discrete interpretations of the stimulus.

As noted in Chapter Five, the meaning of the word gestalt is a little hard to pin down. The concept itself underwent significant evolution, taking on an almost mystical aura in some of the later writings of the Gestalt school. The original meaning was apparently similar to what today would be called a segment. In Köhler's words, a gestalt is "a concrete individual and characteristic entity, existing as something detached and having a shape or form as one of its attributes" [Köhler, 1947]. A more general definition (*e.g.* that of [Koffka, 1935]) would stress the idea of organization into a part/whole hierarchy. That is, a group of features forms a gestalt when each of them has been assigned a role in a common higher-level construct. The constructs may be at a very high level indeed, as in the famous 'duck-rabbit' and 'wife/mother-in-law' sketches; or quite low, as when an array of dots is organized into horizontal or vertical lines. In either case the gestalt organization has two critical properties. The first is discreteness: a given feature either does or does not participate in a given gestalt. There is no middle ground. The second key property is that the organization is in some sense consistent. It satisfies a possibly very complex set of constraints, some hard (*e.g.* a given edge cannot simultaneously be concave and occluding) and some soft (*e.g.* in shape-from-shading situations, the light source is more likely to be above the scene than below it.)

The definition of a gestalt as organization into a part/whole hierarchy has a natural interpretation in terms of Hough transforms. A gestalt simply consists of a winning cell in the output space, plus those features that voted for it. This idea is due to Ballard [Ballard, 1984]. Note however that the Hough transform by itself does not have the requisite discreteness and consistency properties.

Are there other ways to capture the flavor of gestalt organization in a Hough transform network? This chapter presents a mechanism called ***feature binding*** that does the job fairly well. The essence of the mechanism is that the features vote for high-level constructs that they are consistent with, just as in the standard Hough transform. In addition, however, the higher-level constructs compete for the right to explain and receive activation from the features. The winning construct 'binds' the feature and keeps it from contributing to other constructs, whence the name. The binding process obviously yields the discreteness property that is needed. It also provides a means of capturing the constraint that a given feature belongs to only one gestalt, *i.e.* that the part/whole hierarchy is a tree rather than a DAG.

6.1.1 Related Work

One of the claimed strengths of connectionist models is their ability to offer a biologically plausible mechanism for the formation of Gestalts. The general approach was well expressed by Hinton [1981]:

> ...the mechanism underlying the formation of a Gestalt is a set of competitive and cooperative interactions within a network of simple units. The interactions result in a particular subset of the units becoming active and suppressing the rest. The active subset is the internal representation of the current Gestalt.

Hinton's description applies to the obvious implementation of relaxation labelling as well as to specifically connectionist models, and to general networks as well as to the Hough transform nets that are our primary concern. The standard connectionist version of the idea is what [Feldman and Ballard, 1982] call a stable coalition. This is a group of units whose interconnections are all excitatory, and which have inhibitory links to some members of other coalitions. The idea is well illustrated by the Necker cube networks of [Feldman, 1985] and [Rumelhart *et al.*, 1986]. The simplest type of stable coalition is the winner-take-all network, in which a set of units is fully connected by inhibitory links. The result is that the unit with the strongest external input suppresses all other members of the group.

There have been a number of other attempts to represent gestalt-like interactions in connectionist networks. Smolensky's Harmony theory [Smolensky, 1986] captures soft consistency constraints by encoding frequency of co-occurence in the connection strengths of a fully connected single-layer network. The classical literature on relaxation labelling is also highly relevant, as is some of the more recent work on Markov random fields. In the area of Hough transform networks, we have already noted Ballard's [Ballard, 1984] approach to gestalt representation, which is close to that presented here. In his view a gestalt consists of a stable coalition across layers of a Hough transform network, *i.e.* between feature and parameter spaces.

The idea of competition for ownership of evidence, which is the essence of the feature binding, has occurred in a few other contexts. Levitt [Levitt, 1986] builds inference hierarchies representing the probabilities that a given feature belongs to any of several possible explanations. Very recent work by [Califano *et al.*, 1989] is extremely close to the feature binding idea. They formulate object recognition as a hierarchical Hough transform problem, drawing heavily on the approach of Ballard. Instead of restricting the class of transforms, however, they detect cases where two higher level constructs depend on the same features, and allow the constructs in such cases to inhibit each other. Their goal is a practical program rather than a biologically plausible model, so the mechanism is expressed procedurally; if two constructs share more than a certain percentage of their support, then they inhibit each other, otherwise not.

6.2 Feature Binding: The General Approach

Suppose one is given a set of *features* in an image, and wishes to interpret each of them as a manifestation of some underlying *explanation*. For example, one might observe edge segments and try to explain them in terms of lines. The Hough transform approach to the problem is simply to let explanation units add up the values of compatible feature units. The feature binding idea calls for explanation units to compete for the right to explain (and receive activation from) the feature units. Thus explanation units will compete only when their feature sets overlap.

It turns out that a single generic network structure (Figure 6.2) can handle a wide variety of feature binding methods. Each point in feature space has associated with it two groups of units. The first or *bottom-up* group consists of raw features – the output of lower level processing. The second or *top-down* group records information about the set of explanations that are competing for ownership of that point in feature space. Explanation units have a site for every point in feature space that can vote for them. The site function's job is to take into account the competition information and provide appropriate activation to the unit function. It can be thought of as a dynamic weight that is adjusted to reflect the explanation unit's claim on the feature.

6.2.1 A Stable Coalition Approach

Unsurprisingly, a straightforward application of the stable coalition idea can implement feature binding. In the terminology of the previous section, one creates a top-down unit for each possible proposition of the form "Feature x belongs to explanation k." The top-down units for each feature location are placed in a winner-take-all network. The added units are similar in spirit to the binder units of [Feldman, 1982]. Clearly the explanation and associated binder units form rivalrous coalitions. The result is that explanations inhibit each other via the binder units.

The problem with this mechanism is the number of binder units required – one for each possible pairing of a feature with an explanation. Thus the approach must be ruled out for situations in which the features are consistent with many possible explanations. A partial fix would be to associate binder units with whole classes of explanations, rather than specific ones. For example, suppose the feature space consists of lines. It might be enough to have two binder units, one for each possible assignment of figure and ground labels to the regions on either side of the line. This is actually a type of coarse coding, in which specificity of the binder units is sacrificed in return for reasonable network size, in the hope that intersecting a number of coarse constraints will yield a unique solution.

The stable coalition approach has been used by Cooper [1989] to bind scene features to the parts of a structured model for object recognition. He uses a despaced representation of the object to be recognized, so that although each feature has many

possible explanations, the number of features is small. This avoids the combinatoric problems that would arise in more traditional Hough transform applications.

6.2.2 The Rumelhart and McClelland Approach

In an appendix to their well-known paper on past tense learning, Rumelhart and McClelland [1986a] describe a network in which binary features contribute evidence to sets of explanations. They propose to let explanations compute a weighted sum of consistent features, with weights being chosen dynamically so that a) the total activation provided by any given feature is one, and b) the amount of excitation an explanation derives from a given feature is proportional to the current activation of that explanation. Extending their approach to networks with weights produces a powerful and effective feature binding rule, as follows:

Let F_x be a set of features, and let the activation of explanations E_k be determined by

$$E_k := \sum_{x \in I_k} w_{xk} g_{xk} F_x. \tag{6.1}$$

where w_{xk} is a static evidential weight and the set I_k contains the indices of all features which provide evidence for E_k. The term g_{xk} is a dynamic "gating" weight whose value is between 0 and 1. It can be thought of as a switch or valve controlling how much activation flows from F_x to E_k. The gating weight is computed by the formula

$$g_{xk} = \frac{E_k}{\sum_{j \in O_x} E_j} \tag{6.2}$$

where the set O_x contains indices of all explanations for which F_x provides evidence. Clearly the g_{xk} for any given x sum to one and are proportional to the corresponding E_k, so the expression specializes to the Rumelhart and McClelland case when the w_{xk} are all equal to one.

It is not hard to map the above expressions onto the generic network of figure 6.2. Note that the denominator term $\sum_{j \in O_x} E_j$ is independent of k; call it B_x. Intuitively, B_x is the back-projection of the relevant explanations to feature F_x. Let F_x and B_x be, respectively, the bottom-up and top-down feature units. The site function for any site x on explanation unit k can then compute g_{xk} (assuming it has access to E_k) and use it to weight $w_{xk}F_x$. The explanation unit simply computes the sum of its sites.

6.2.3 An Example: Organizing Dots into Lines

A simple example will demonstrate the use of the feature binding technique and some of its behavior. Suppose that we are given an array of dots and wish to organize them into horizontal or vertical lines. Figure 6.3 a) shows a feature binding network for this problem. The network contains two rectangular arrays, one for the dots (the F_x

units) and one for the feedback units B_x. Both provide input to feature binding sites on the units that code for lines. Connections are shown for the input and feedback units at location (6,4) and line units ($y = 4$) and ($x = 6$). Input units have their values externally clamped, and feedback units compute an unweighted sum. The line units will normally use the activation rule of expressions 6.1 and 6.2 above, but first some practical problems must be solved.

First, assume that the B_x and line units are updated simultaneously at the beginning of every cycle. Then the B_x units will reflect the state of the E_k units during the previous cycle. If g_{xk} is computed using the current values of E_k and B_x, the feedback delay may make the network oscillate. In the simulations described below the E_k units avoid the problem by using a delayed copy of their own activation to compute g_{xk}. The second problem is that g_{xk} is not defined when the feedback units have value zero. This situation may occur whenever a new input unit is turned on. When this happens, however, the explanation units that use the feedback unit in question will be zero as well. Since under normal conditions the g_{xk} for any given x sum to one, it seems reasonable to define g_{xk} to be $1/|O_x|$ when B_x is zero.

Figure 6.3 b) shows the output of the network of Figure 6.3 a) given an ambiguous set of dots. Since the features provide exactly equal evidence for both interpretations, the algorithm cannot choose one over the other[1]. Figure 6.3 c) shows what happens when the evidence favors the horizontal interpretation by a small amount. Note that the interpretation is globally consistent: strengthening the ($y = 2$) line weakens the two vertical lines, allowing the ($y = 5$) line to win even though nothing explicit was done to favor it.

6.2.4 Analysis

We can obtain a clearer understanding of what the algorithm computes by viewing it as a relaxation labelling problem in which we features are labelled with explanations. In particular, it can be mapped into the formalism of Hummel and Zucker [1983]. We are given

a set of objects — The features.

a label set for each object — The dynamic weights g_{xk}. The value of g_{xk} can be thought of as 'the strength of label k at feature x'.

a neighbor relation over objects — Two features F_x and F_y are neighbors iff they have any possible explanations in common - that is, if $O_x \bigcap O_y$ is non-empty.

[1]It can be shown that if random noise is added to the weights, this situation always converges to an unambiguous interpretation. Unfortunately the convergence rate may be arbitrarily slow.

a constraint relation over tuples of neighbors — Define $r_{xy}(k,j)$, the compatibility of label k at x with label j at y, to be

$$r_{xy}(k,j) = \begin{cases} w_{xk} & \text{if } k = j \\ 0 & \text{otherwise} \end{cases}$$

Under this mapping Hummel and Zucker show that if the g_{xk} are updated according to the rule described in [Mohammed *et al.*, 1983] (henceforth the MHZ rule), then one of two cases will apply. If the static weights w_{xk} depend only on k, as in typical Hough transform applications, then the compatibility matrix is symmetric, and the algorithm performs gradient ascent on the functional $\sum E_k$. This means that it is guaranteed to halt, and will tend to maximize activity in the output network. If the w_{xk} depend on x as well as k then convergence can no longer be proven. However, Hummel and Zucker give an argument that convergence is in some sense probable.

The next question to ask is whether feature binding in fact implements the correct update rule. The answer, unfortunately, is no, but it does approximate it in many situations. In order to see this it is necessary to look at the update rule in some detail. Figure 6.4 a) depicts the situation geometrically for one node having three possible labels. Since $g_{xk} > 0$ and $\sum_{\text{all } k} g_{xk} = 1$, the set of valid labellings is just the positive part of the n-dimensional plane given by $(g_{x1}, \ldots, g_{xn}) \cdot (1, \ldots, 1) = 1$. The MHZ update rule says that where possible[2], the new labelling should be computed by taking a small step along the projection of the support vector $(E_1, \ldots, E_n)$ onto the plane of the solution set. Algebraically, the rule can be written as

$$g_{xk} := g_{xk} + \alpha \left(E_k - \frac{1}{|O_x|} B_x \right). \tag{6.3}$$

where α controls the step size. The effect of expression 6.3 is to modify the support vector by subtracting away its projection onto the plane normal, yielding a vector that lies in the plane. Figure 6.4 b) illustrates this graphically.

The feature binding update rule can be written as

$$g_{xk} := g_{xk} + \frac{1}{B_x} (E_k - g_{xk} B_x), \tag{6.4}$$

corresponding to the situation in Figure 6.4c). The similarity to expression 6.3 is clear. The one important difference is that the projection is done parallel to the current label vector rather than the plane normal. Early in the relaxation the label vector will be near the plane normal. As long as the support vector is *not* near the plane normal, feature binding will approximate the correct update rule very well. When these conditions do not hold, however, situations can arise in which the feature binding rule moves the label vector in exactly the wrong direction. Figure 6.4 d) illustrates such a situation for a two-label network.

[2]Special rules apply when the current labelling is on a boundary of the label set and the projection of the support vector points away from the label set – see [Mohammed *et al.*, 1983] for discussion.

6.2.5 Improving the Update Rule

The feature binding update rule works reasonably well for many applications, despite its occasionally poor approximation to the correct rule. Given that the problem does exist, however, we would do well to look at ways of improving the situation. The simplest approach would be to modify the support function. Suppose each term in the expression for g_{xk} is exponentiated, *i.e.*

$$g_{xk} = \frac{|E_k|^h}{\sum_{j \in O_x} |E_j|^h} \tag{6.5}$$

Let M' be the set $\{i : E_i = \max_{j \in O_x} E_j\}$: then as h goes toward infinity, g_{xk} approaches

$$g_{xk} = \begin{cases} |M_x|^{-1} & \text{if } k \in M_x \\ 0 & \text{otherwise} \end{cases}$$

and the largest E_k get all of the activity due to F_x. In terms of Figure 6.4, the effect is to pull the support vector toward the corners of the label space and away from the plane normal. This in turn will tend to make the update rule give a better approximation to the correct rule. Note however that the support function will no longer be describable by a simple compatibility matrix.

A more sophisticated approach is to try to implement the MHZ rule directly. It would be relatively trivial to implement the basic update rule of expression 6.3. The B_x units would simply scale themselves by the constant $|O_x|^{-1}$, and the E_k units would update g_{xk} appropriately at each site. This might cause g_{xk} to become greater than one or less than zero. This could be handled by simple truncation, though that strategy might lead to solutions that fail to satisfy the constraint that the g_{xk} should sum to one.

Somewhat surprisingly, it is possible to implement the full MHZ update algorithm without violating the connectionist design rules of [Feldman and Ballard, 1982] or requiring a unit for every label. The unit functions required strain our intuitive notion of simplicity, so the result may not be terribly relevant to biological models; but it is interesting to see what can be done when the formalism is pushed to the limit.

Figure 6.1 presents the MHZ algorithm for computing the update vector, restated using the notation of this chapter. The algorithm uses the current label vector $\vec{G} = (g_{x1}, \ldots, g_{xn})$ and the current support vector $\vec{E} = (E_1, \ldots, E_n)$ to compute a new update vector $\vec{U}$. The new vector is chosen to maximize $\vec{E} \cdot \vec{U}$ subject to the constraint that $|\vec{U}| \leq 1$ and $\vec{U}$ is a feasible direction (*i.e.* $\vec{G} + \alpha\vec{U}$ is in the set of legal labellings for all α between 0 and some positive constant). To map this onto the connection network of Figure 6.2, we let the E_k units update themselves and the g_{xk} according to expression 6.3. The B_x units become considerably more complicated. At each input site k they must store a copy of g_{xk} and information about whether k is in set T;

Input $\vec{G}$ and $\vec{E}$:
Set $T = \emptyset$:
repeat
$B_x := \frac{1}{n-|T|} \sum_{i \notin T} E_i$
$T := \{i \mid (g_{xi} = 0) \wedge (E_i < B_x)\}$
until T is stable:

$$y_i = \begin{cases} 0, & i \in T \\ E_i - B_x & \text{otherwise} \end{cases}$$

$$u_i = \begin{cases} \vec{y}, & |\vec{y}| = 0 \\ \vec{y}/|\vec{y}| & \text{otherwise} \end{cases}$$

return $\vec{u}$

Figure 6.1: Pseudocode for the MHZ update direction computation.

during each time slice they must compute the new value of B_x and update their local copies of g_{xk} by implementing the **repeat** loop of Figure 6.1. Both the explanation and feedback units must either use a very small α or choose α dynamically to prevent the g_{xk} from moving outside the interval $[0, 1]$.

6.3 Feature Binding and the Hough Transform

The classical Hough transform is subject to a number of difficulties that can be at least partially solved by feature binding. In the Hough transform features contribute activity to each higher level interpretation of which they might be part. Interpretations which have a lot of supporting features will have higher activity, but one is left with the problem of deciding whether a given level of activity at an explanation indicates something real or is simply the result of an accidental alignment of votes from unrelated features (cf [Brown, 1983b].) Feature binding causes features which are contributing to a strong explanation to contribute correspondingly less activity to other explanations. We would expect therefor that it would reduce the noise level and simplify the thresholding problem.

Another classic problem with the Hough transform arises due to interactions between input noise and output quantization. In most applications the features are the result of some measurement process that is subject to noise and error. In the line-finding case, for example, edge detectors are subject to position and orientation errors. If the cells of the output parameter space are finely quantized, as they should be to give high resolution, then the input errors will produce a blurred hill in the vote array rather than a sharp peak. Thresholding the vote array to find lines will

typically result in finding a cluster of similar lines for each real line in the image. The usual heuristic applied in this case is some form of lateral inhibition in the output space. That is, local maxima are extracted and allowed to suppress their neighbors over some local region. In many situations, however, this does not make sense. Consider Figure 6.5 a), which shows the image representations of two lines taken from a line-finding Hough space. Should line L_1 inhibit L_2? The answer is that it depends on the input. If the input looks like Fig. 6.5 b), there is independent support for each line, and inhibition would clearly be a mistake. If on the other hand the input looks like Fig. 6.5 c), the votes for L_1 and L_2 are likely due to noisy measurements, so inhibition would be appropriate. Feature binding handles this problem nicely if the noise properties of the features are known. Each feature casts a distribution of votes and is competed for by all of the solutions it votes for. Two solutions will inhibit each other if and only if they depend on common features.

There are of course a variety of other ways of dealing with the Hough transform's problems. A standard 'folk algorithm' for dealing with the noisiness and ambiguity of the output space is to find the peaks sequentially, deleting their features along the way. That is, after voting one repeatedly finds the output cell with the highest vote count, adds it to the list of solutions, and deletes all of the features that are consistent with it. This idea is quite similar in spirit to feature binding, and one might well ask whether feature binding has any advantages over it other than being inherently parallel. In fact feature binding computes a different set of solutions, and in some cases the solutions it finds seem more natural. Consider the network of Figure 6.6, which finds adjacent pairs of dots[3]. Since the weights to the middle dot-pair unit are higher than those to the outer units, initially the middle output unit will receive the most votes. The maximum-subtraction heuristic will choose it first, and be left with weak activation on the other two units. Feature binding, on the other hand, will settle into a state in which the middle dot-pair unit is suppressed and the outer two units are fully active.

Another way to reduce ambiguity in the Hough space is to use pairs of features to generate votes. In connection networks this translates into the use of conjunctive connections on the output units. The strategy works well, and has been elevated by Ballard (*e.g.* in [Ballard, 1984]) to a fundamental design principle. It does not, however, deal with fundamentally ambiguous stimuli like those of Figure 6.3 b).

The problem of using noisy input with small accumulator bins can be dealt with by using large bins and interpolating to improve spatial resolution – *i.e.* the interpolation coding technique mentioned in Chapter Five [Ballard, 1986c]. This also has the advantage of reducing the number of units needed to represent the output space. Like all distributed representations, of course, the method can become confused if a number of peaks occur close together. A subtler problem is that the method requires features to vote for points in the output space rather than surfaces. This is one of the

[3]Readers in search of a use for this may think of the output units as line segment detectors.

motivations for Ballard's reliance on conjuctive connections [Ballard, 1984]. It also imposes streng constraints on how the transformation from feature to output space is computed. It may be necessary to break the transform down into a hierarchy of steps, each of which maps a pair of features and one level to a single point at the next level.

6.4 Extensions and Variations

In this section we will look at a number of ways of altering the behavior of feature binding networks to get different sorts of behavior or handle more general problems.

6.4.1 Biasing Feature Binding Networks

When the Hough transform is used in a connectionist context it is often necessary to have ways to bias the network so that its decision takes into account external biases as well as the evidence provided by the actual features. In standard winner-take-all networks this is done by simply adding activation to favored units. In a feature binding network, however, additive bias has effects that may be undesirable. In a feature binding network the solution units compute a weighted sum of feature inputs, but compete for ownership of each feature. Additive bias looks like a feature for which there is no competition – a feature which is specific to one particular solution. It has the effect of putting a floor under the activation of that solution. If the solution in question does not actually win the competition, therefor, the network will converge to state in which multiple inconsistent interpretations are active.

It turns out that in most cases more intuitively satisfying results can be obtained by using multiplicative rather than additive bias with feature binding. That is, for bias inputs b_j the explanation unit activation rule becomes

$$E_k := \prod_j b_j \sum_{x \in I_k} w_{xk} g_{xk} F_x, \tag{6.6}$$

where values of b_j greater than one denote positive bias, values less than one negative bias. It should be obvious that this is exactly equivalent to multiplying all of E_k's input weights (the w_{xk}) by the product of the biases. Thus its effect is to modulate the unit's sensitivity to feature input.

Another type of bias that one might wish to apply would be to alter the relative importance of some features, making them more or less salient. Simply changing the activation level of the features in question would work in some situations. In others the feature may exert its effect via some complicated site function, or may encode information it its activity (as in the case of the decay clocks introduced in the last chapter.) In these situations a more subtle method is needed. What can be done

in such cases is to divide the bias into the feedback unit associated with the favored feature. That feedback unit normally insures that the dynamic weights from that unit to its explanations sum to one. Dividing the feedback unit by a bias term $b > 1$ allows the sum of the weights to rise to b, increasing the impact of the associated feature on the network's solution. Where the previous bias rule rescaled all of the weights *in* to some E_k, this rule is equivalent to multiplying all of the weights *out* of F_x by b.

6.4.2 Feature Binding with Interpolation Coding

As stated earlier, Ballard's interpolation coding provides some of the same benefits as feature binding, but there are still good reasons to try to add feature binding to an interpolation coded network. A network using both techniques would inherit feature binding's ability to handle fundamentally ambiguous inputs like that of Figure 6.3 b), while still retaining the efficiency and high resolution of interpolation coding.

Consider the simplest version of interpolation coding, in which some feature F_x votes for an ideal point E_k in a one-dimensional parameter space. (The approach generalizes trivially to higher dimensions.) Suppose that the output space does not contain a unit for E_k, but instead represents it by interpolation between two values E_1 and E_2. Following [Ballard, 1986c], we link F_x to both E_1 and E_2 with static weights

$$w_{x1} = \frac{E_2 - E_k}{E_2 - E_1} \qquad \text{and} \qquad w_{x2} = \frac{E_k - E_1}{E_2 - E_1}.$$

If feature binding is applied in the usual way, the result of turning on F_x will be a competition between E_1 and E_2, which will be won by whichever is closer to E_k. To avoid this, E_1 and E_2 must be made to compete as a unit against other pairs. This can be done by making connections between them and letting $g_{x1} = g_{x2} = (E_1 + E_2)/B_x$. Since the two explanation units now have identical dynamic weights, their relative activity (which is what encodes the precise location of E_k) will not be distorted by the feature binding process.

6.4.3 Feature Binding in Hierarchical Networks

In connectionist models of perception it is common to arrange a series of Hough transform-like computations in a hierarchy, with the output cells at one level playing the role of features at the next level [Ballard, 1984; Feldman, 1985]. We would like to be able to use feature binding in this type of situation. In particular it would be good to have a way for higher levels of the network to influence the relaxation at lower levels. In that way ambiguities at lower levels might be resolved by more global considerations. Such a technique would also allow the use of Ballard's 'loosely coupled subspace' idea (see Chapter Five) for reducing the size of Hough spaces.

The desired behavior can be obtained by application of the biasing methods presented above. Figure 6.7 shows the type of network needed. Each layer of the hierarchy has three classes of units:

- **representation units**
  These are the skeleton of the network. They represent the parameter space, playing the role of E_k units relative to the layer below, F_x units relative to the layer above.

- **feedback sum units**
  These are the B_x units previously discussed. They record the total activity of all explanations for each unit at this level (in its role as a feature).

- **feedback max units**
  These units, which will be denoted by M_x, are exactly like B_x units except that they compute the max over their inputs rather than the sum. That is, they record the activation of the strongest explanation for each feature.

Each representation unit will use one plus the value of the associated max unit as a bias, i.e.

$$E_k := (1 + M_k) \prod_j b_j \sum_{x \in I_k} w_{xk} g_{xk} F_x. \tag{6.7}$$

Early in the relaxation the higher levels of the network will be in a highly ambiguous state, with low levels of activity distributed broadly across the units. Under these conditions the top-down bias will have little or no effect. Only when the higher levels have enough information to settle on at most a few candidate explanations will they begin to have a significant impact on what happens at lower levels. Feature binding networks settle very quickly when the input is unambiguous, so in those cases top-down feedback is unlikely to be available in time to make a difference. The result is that top-down feedback will be invoked only when the input to the lower level is highly ambiguous.

6.5 Conclusion

We have argued that gestalt percepts can be usefully modelled as the result of a Hough transform-like computation in which units at the output side of each transform compete to receive activation from units at the input side. This idea can be implemented in a connectionist network with only a modest increase in the resources required, though it does significantly complicate the activation functions of the units. We have devoted most of our attention to a particular activation function based on an idea of Rumelhart and McClelland. The rule can be analyzed as an attempt to label features with explanations using the relaxation mechanism of Hummel and Zucker. It

can also be extended to incorporate biases, and to the case of hierarchies of transform networks.

In the next chapter we will combine feature binding with some of the techniques described in the previous chapter to build a network that analyses motion sequences.

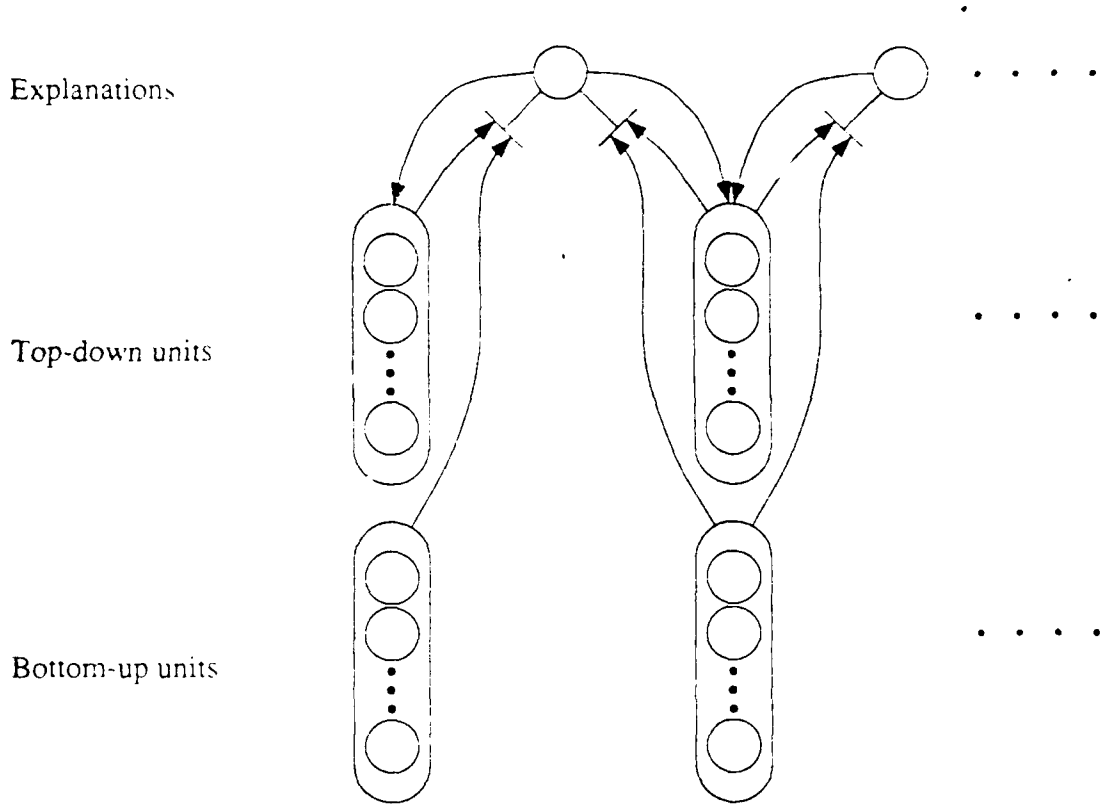

Figure 6.2: generic feature binding network.

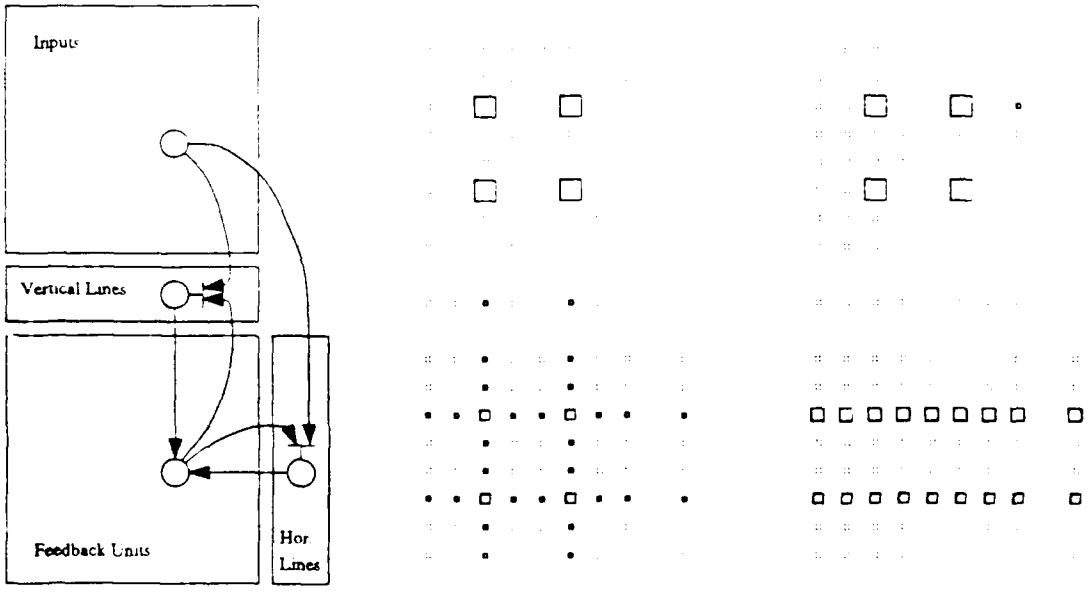

Figure 6.3: a) (left) A network that organizes dots into horizontal and vertical lines. b) (center) Stable state of the network given an ambiguous stimulus. c) (right) Stable state of the network given an unambiguous stimulus.

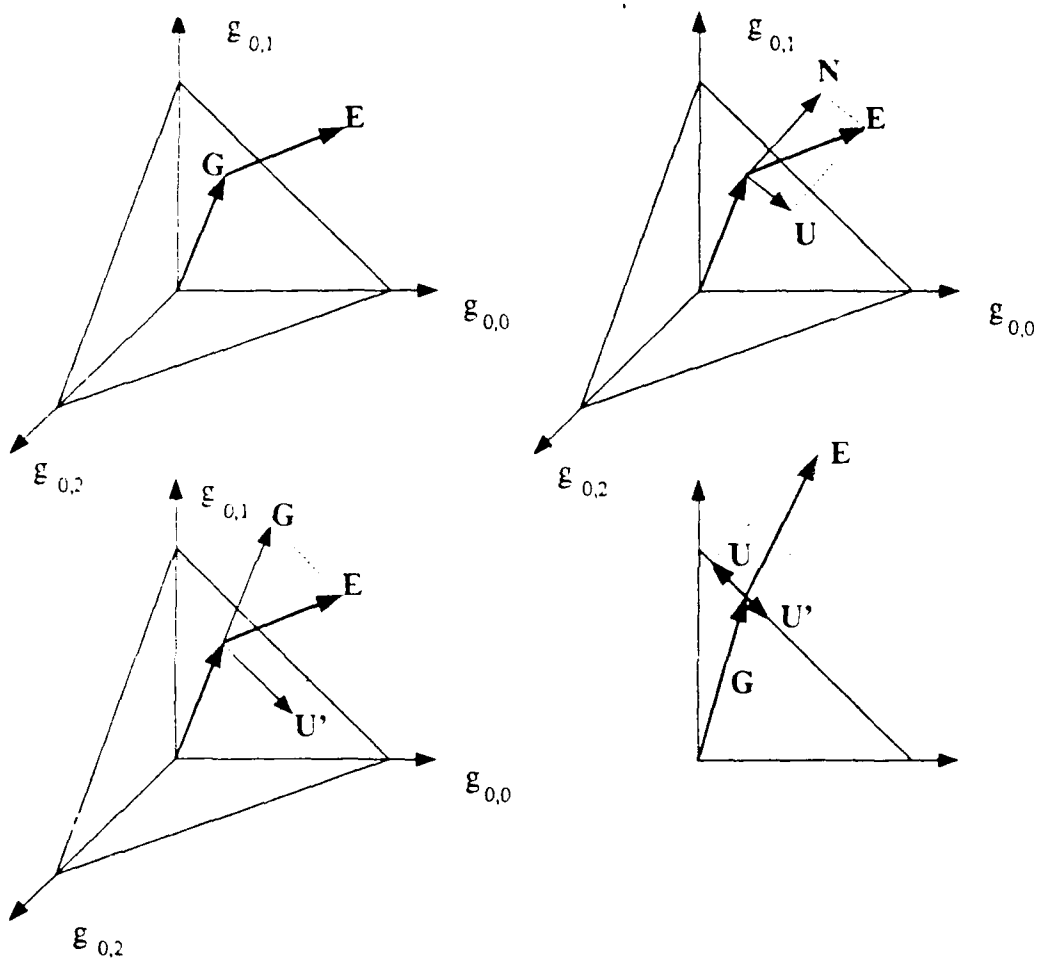

Figure 6.4: Geometric interpretation of feature binding as relaxation labelling. a) The label space for a single node with three possible labels. b) The MHZ algorithm projects the support vector parallel to the plane normal. c) Feature binding projects parallel to the current label vector. d) If the support vector lies between the plane normal and the current label vector, feature binding will update the label vector incorrectly.

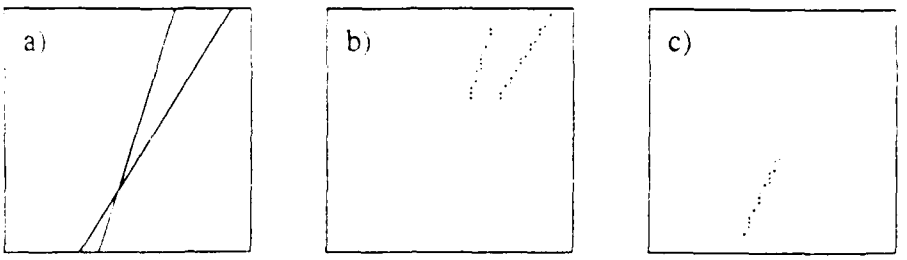

Figure 6.5: A case where lateral inhibition in Hough space fails. a) lines represented by two nearby cells in Hough space. b) an image for which the lines should not inhibit each other. c) an image for which they should.

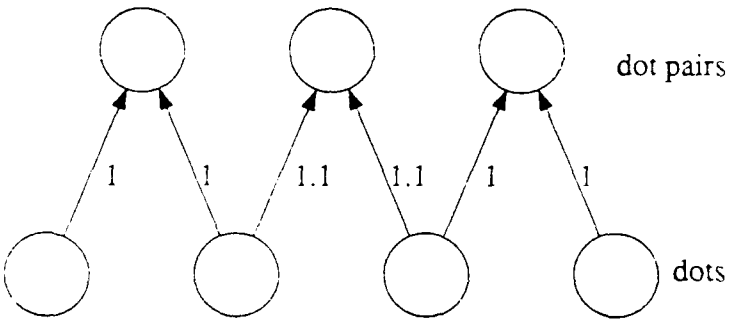

Figure 6.6: A network that finds adjacent dot pairs.

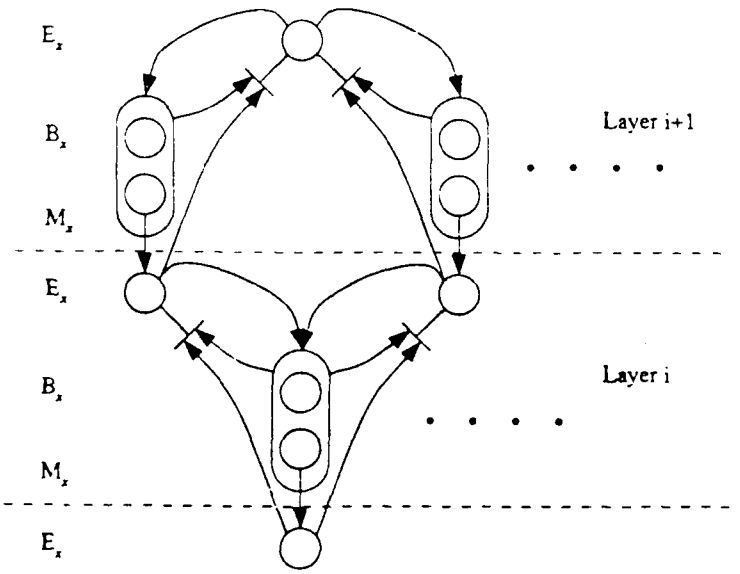

Figure 6.7: A network for hierarchical feature binding.

7 A Network for Intermediate-Level Motion Understanding

Chapters Five and Six described techniques for representing temporal information and gestalt interactions in connectionist networks. In this chapter those techniques will be used to implement a model of intermediate-level motion processing along the lines sketched in Chapters Four and Five. This will involve committing to particular representations for trajectories, segments and time, rather than talking in general terms as in previous chapters. As stated earlier, the particular representations used here may well turn out to be incorrect, in the sense that they differ from what the brain uses. This is unavoidable given the state of current knowledge. Despite this the results should be of interest to anyone interested in how the brain analyses motion. First, they demonstrate the types of problems that any motion processing system will encounter. Second, they provide an existence proof that the architecture sketched in Chapter Four is computationally practical.

The next section presents a precise statement of what the domain of the network will be and what sort of output it should produce. Section 7.2 discusses the basic design of the network, specifying the representations, activation rules and connection patterns needed to interpret basic stimuli. In subsequent sections the network will be augmented with mechanisms that let it handle more complex effects. The chapter closes with a look at how the network might be extended to things that are presently beyond its scope.

7.1 The Model Domain

As stated at the end of Chapter Five, this thesis is not intended to address the question of how the visual system solves the segmentation problem. In order to model intermediate-level motion, therefor, it will be necessary to restrict the model to a domain in which segmentation can be assumed. Working in restricted domains or 'toy worlds' is of course a standard practice in computer vision, though we are sympathetic

to recent criticisms that the approach can lead to a false sense of confidence in the effectiveness of our algorithms (*e.g.* [Brooks, 1987]).

The properties that a reasonable toy world should have are as follows. First, it should cover a reasonably rich and interesting set of visual stimuli. The set should be varied enough to allow testing of all of the capabilities of the model described in chapter Five. Second, it should avoid raising any secondary issues, and should be no more complex than necessary. In particular, it should avoid the segmentation problem.

The domain of the model will be a two-dimensional world of moving blobs. Blobs will be assumed to have been abstracted by some segmentation process into property vectors at discrete locations; for example, the input might represent assertions of the form "there is a red, 1° circle at location (2,7)". The blobs will rotate and translate along straight or circular trajectories at constant speeds. The network's job will be to recover the trajectories and speeds of the blobs, and it will be expected to work regardless of temporal sampling rate. Thus the domain includes almost all classical apparent motion displays, plus a non-trivial set of continuous motion stimuli.

7.2 Designing the Network

The motion network will be an elaboration of the design sketched in section 5.3 and figure 5.1. It will have the general character of a Hough transform, in which the property vectors representing blobs vote into a space of trajectories. The basic Hough paradigm will be modified in a number of ways, however. Units that encode stimulus age will be added to the blob descriptors, and the trajectory units will use one of the mechanisms described in Chapter Five to take into account the relative ages of local and remote stimuli. The trajectory units will also compete with each other for ownership of the stimuli using the feature binding technique described in Chapter Six. This will help the network arrive at unambiguous interpretations even when the stimuli are ambiguous.

The sections that follow will describe the network at a relatively painful level of detail. The intent is that the results presented at the end of the chapter should (at least in principle) be replicatable by other vision researchers. It is not meant to imply that details such as the particular way in which parameters are quantized are intrinsic features of the model. The model should be judged first on its performance, and second on whether it appears possible to extend it to a full range of motion phenomena.

7.2.1 The Input Space

The input to the network will be an 8×8 spatially indexed array of blob descriptors. It is intended to represent the output of the segmentation process, and to correspond

to the iconic array shown in figure 5.1. The array will contain the following units or unit groups:

- **stimulus present**
  Each descriptor will have a single stimulus present unit that is on when there is currently a blob at its location, off when there is not.

- **categorical property**
  Each location will have a group of units that encodes any blob properties that can be compared by simple binary or scalar measures. In the present implementation these consist of three units called **red**, **green** and **blue**. They should not be interpreted as literally representing color, however, but rather as arbitrary properties to be used to compute the similarity of two segments.

- **complex property**
  Not all properties can be compared by simple scalar measures. In the paradigmatic example of shape, useful comparisons will have complex parametric descriptions, such as "these shapes match with strength s under rigid transformation T". In order to show how such properties can be used, each location will have a set of units describing local stimulus shape.

- **clock**
  The time since onset of the local stimulus will be encoded by a linear decay clock of the type described in Chapter Five. The clock will be triggered whenever it detects an off-to-on transition in the associated stimulus-present unit.

The stimulus present and categorical property units will be binary (outputs either zero or one). Shape property units will use a level encoding scheme to be described later. The clock units will produce an output of 50 when triggered, and will decay at a rate of 1 per time step. All but the clock units will be set externally by whatever process presents the stimulus to the network.

At stimulus onset, the stimulus present and property units will be turned on and the clock will be triggered. The stimulus present unit will be turned off at stimulus offset, but the property units will remain active as long as the clock continues to run. If a new stimulus arrives before the clock has expired, the old clock and property information will be overwritten.

7.2.2 Trajectory Space

The trajectory space is intended to encode all line and arc segments at constant speeds. As observed in Chapter Five, the obvious parameterization would require CSn^4 units for C curvatures, S speeds, and an $n \times n$ array. In the current design $n = 8$, and curvature and speed will be coarsely quantized, so a full unit value

representation would not be out of the question. However, it would almost certainly be impractical for more realistic values of n. Chapter Five described a number of ways of recoding parameter spaces to reduce their size. The strategy adopted here will be a combination of subspace decomposition and interpolation coding. Separate parameter spaces will be used to represent the starting and ending points of each trajectory. The subspaces will be partially coupled by addition of a parameter that encodes the local tangent direction. That is, trajectory Start and End (SE) units will be parameterized by location, local tangent direction, curvature and speed. Figure 7.1 shows a sample ideal trajectory and its representation in terms of SE units. Trajectory speed will be handled by an interpolation coding technique. For each location, direction and curvature there will be two units, coding for FAST and SLOW speeds. The speed of a particular trajectory will be represented by the relative activation of the two units.

Local tangent direction will be quantized into the eight values making angles of $n\pi/4$ with the x axis. Curvature will be quantized into five canonical values corresponding to straight lines and arcs of radius ± 5.3 and ± 2.7[1] Thus each trajectory unit will stand for trajectories lying within a region having the form of a (possibly bent) cone. Figure 7.2 shows the cone-shaped regions for the units at one location whose direction parameter specifies horizontal motion to the right. In the figure the squares that make up each cone represent locations that are linked to the corresponding unit. Thus the cone-shaped regions can be though of as receptive fields. Because of the temporal sensitivity of the site functions, the receptive fields are oriented in space-time in the manner of the spatiotemporal energy models discussed in Chapter Two. They differ, however, in that they operate on blob descriptors rather than contrast patterns, and in that they code for particular trajectories rather than normal optical flow.

A Note on Scaling

While on the subject of representing the parameter spaces it is worth looking at how the design would scale to more realistic input array sizes. For D local tangent directions, C curvatures, and S speeds the current parameterization requires $DCSn^2$ Start units and an equal number of End units. The current implementation uses five curvatures, eight directions and two speeds with an 8×8 input array, for a total of about ten thousand SE units. Expanding the input array to 100×100 would raise the total to 1.6 million units, still small by biological standards. In practice the number of directions and curvatures represented would have to increase somewhat with increasing array size, so that the asymptotic complexity of the network is between $O(n^2)$ and $O(n^3)$. Even so, the number of units required scales reasonably well. For example, 10 curvatures, 32 local directions and 4 speeds on a 100×100 array requires only about 26 million units.

[1] Lengths are in units defined by the spacing of the input array. That is, neighbors in the input array are one unit apart, et cetera.

In fact it is unlikely that the parameter space would need to be even this large. According to the architectural design sketched in Chapter Four, the intermediate-level vision system uses an 'attentional' coordinate system whose mapping to retinal coordinates is programmable. If the mapping can include scaling transformations as well as translations, then the intermediate level can get by with a relatively small number of distinct spatial locations. When an observer attends to a small region of retinotopic space, the attended region will be scaled to cover the whole intermediate-level array, and he will have good spatial resolution over a small part of the scene. When he attends to the whole scene the scaling factor will be much smaller, and he will have low resolution over a much larger area.

Trajectory Unit Computation

The trajectory unit activation function is where feature binding and temporally sensitive computations come into play. Understanding it is the key to understanding how the network behaves. The trajectory units will use a variant of the activation rule given in equation 6.6:

$$E_k := \prod_j b_j \sum_{x \in I_k} w_{xk} g_{xk} F_x$$

Here k should be understood to range over all SE units, and x over all locations. The mapping from rule to network can be summarized as follows: The features F_x record, for each trajectory Start or End unit, the extent to which the relative ages of local and remote stimuli are consistent with that trajectory. The static weights w_{xk} encode the *a priori* goodness of correspondence between local and remote locations. They are used to capture things like preference for nearest neighbors and straight trajectories. The dynamic weights g_{xk} are used to let trajectory units take into account the relative strength of other possible explanations for a stimulus, as described in Chapter Six. All three factors (temporal goodness-of-fit, static weight and dynamic weight) are combined at the site level. The trajectory unit sums the activity of each of its sites. Finally, the bias inputs b_j are used to incorporate bias from attention, retinal slip et cetera.

We will now look in detail at how each term in the above activation function is computed.

Temporal Sensitivity Consider a trajectory Start or End unit parameterized by the tuple (X, Y, C, D, S) (that is, location, curvature, local tangent direction, and speed). Suppose it is to be connected it to a clock unit at some location (x, y) in its receptive field. Given the two locations it is easy to compute the length of the unique arc segment that joins them and has the appropriate local tangent direction. That length and the speed S in turn specify the temporal asynchrony T_{opt} that is optimal for that connection.

Since T_{opt} depends only on static parameters, it can be precomputed and built into the site function for each connection. The precise rule that the sites use to compute temporal goodness of fit is $F_x = G(t_2 - t_1 - T_{opt})$, where the envelope function G is a triangular pulse. Trajectory speed is interpolated between two canonical values. Figure 7.3 shows how the envelope functions for those two values are related. Speeds between the canonical values are interpolated by the obvious linear weighting function. Speeds outside that range will not be precisely recoverable, although motion may still be detected.

It is well known that negative interstimulus intervals weaken the apparent motion percept. In the network this is accomplished by gating some of the sites by the stimulus present units associated with the locations being compared. In particular, sites that are attempting to interpret some location as the start of a trajectory are disabled by the location's stimulus present unit. The result is that a stimulus pair with a negative ISI will not be put in correspondence until the first stimulus has disappeared. If there are other candidates for correspondence available when the second stimulus appears, the network will begin to settle on one of them. By the time the first stimulus disappears and becomes available for correspondence, it will be at a competitive disadvantage with respect to the other candidates. Whether it will be able to overcome its rivals' head start will depend on how intrinsically plausible it and they are, how long the negative ISI was, et cetera.

Static Weights The static weights are a product of two terms, one penalizing arc length and the other curvature. The length term has value one at an arc distance of zero, and falls linearly to zero at an arc distance of twelve. The curvature term is highest for straight trajectories and falls linearly with the canonical curvature for the trajectory unit in question. Specifically, it is 1.0 for straight trajectories, 0.9 for shallow curves, and 0.8 for sharp curves. One could easily imagine incorporating other *a priori* biases, such as a preference for straight trajectories that are vertical or horizontal over diagonal, or for concave down over concave up, but this has not been done in the current implementation.

Dynamic Weights The feature binding mechanism described in Chapter Six must be elaborated significantly to take account of the differences between the network as described and more standard Hough transform applications. The first problem arises because of the temporally sensitive sites. The activation that a given trajectory unit receives from a given remote location depends on the relative ages of the local and remote stimuli. Thus the temporally sensitive sites are actually a type of conjunctive connection, in that the activation to be derived from one input depends nonlinearly on the value of another input. Dealing with general conjunctive connections in a feature binding network is quite difficult, though it can be done. A somewhat simpler approach works here, due to the fact that the local stimulus appears in all of the

conjuncts.

The intuition behind conjunctive feature binding is that if a given trajectory is losing the competition for a remote stimulus, then it should not be allowed to win locally either, even if it is the best local solution. In other words, it should be allowed to make use of a conjunct only to the extent that it owns both of its terms. This behavior can be obtained by using a dynamic weight that is the *product* of the normal feature binding weights for the individual terms. That is, if a trajectory E_k depends on locations x and y, the dynamic weight will be $g_{xk}g_{yk}$.

The other problem with feature binding in this application is that the representation of the output parameter space (the E_k units, in the terminology of Chapter Six) uses a distributed encoding scheme. It now takes four units to represent a single trajectory: two speed units for the trajectory Start and two more for the End. Clearly the four units representing a single trajectory should not compete against each other for ownership of a common stimulus location. The speed units can be kept from competing by the general method for feature binding with interpolation coding presented in Chapter Six. That is, each pair of units coding for a given location, direction and curvature uses the total activity of the pair as the numerator of the expression for g_{xk}:

$$g_{xk} = \frac{\sum_{\text{all speeds}} E_k}{\sum_{j \in O_x} \sum_{\text{all speeds}} E_j}$$

A different method is required to keep the Start and End units from competing with each other. The key observation here is that when a given feature can play multiple roles in a feature binding network, competition for each role should be mediated by a separate feedback unit. In the current case, a given blob can play any of four roles with respect to the trajectory units: it can be either the local or remote half of a temporally sensitive connection on either a Start or an End unit. For example, the blob at the end of a simple trajectory fills both the 'local end' role for the local End unit and the 'remote end' role for the Start unit at the beginning of the trajectory. These roles are not mutually exclusive, so each will have its own feedback (B_x) unit. The 'local end' feedback unit makes all End units at the location compete, and the 'remote end' feedback unit does the same for remote Start units. 'Local start' and 'remote start' roles also get their own feedback units, since there is nothing inconsistent about a given blob being simultaneously the end of one trajectory and the start of another. The result is that the stimulus can be shared in logically consistent or identity preserving ways. Interpretations in which dots split or merge, however, will be strongly inhibited.

Breaking the competition down by feature roles does not quite solve all the problems, unfortunately. Consider what happens during the processing of a continuous motion stimulus, in which a segment moves through several locations in succession. If the network interprets the stimulus correctly, one would expect to see one Start unit active at the oldest location, one End unit active at the most recent, and no

units active in between. That means that there will be no competition inhibiting the local Start or End units at intermediate locations. Thus if other, unrelated stimuli appear, the intermediate units of the continuous trajectory will try to act as Starts in trajectories involving the new stimuli, defying the intuition that the intermediate locations already have a satisfactory explanation. To prevent this, Start units inhibit other Start units at locations in their receptive fields. They do this by scaling up the activation of the feedback units for the Local Start role at those locations. This has the effect of making those start units less sensitive to input, without entirely preventing them from competing.

Figure 7.4 shows the units and connections for a single pair of start/end units, and Figure 7.5 summarizes the trajectory computation in as compact a form as possible.

7.2.3 A Note on the Implementation

The network described above was implemented using version 4.1 of the Rochester Connectionist Simulator [Goddard *et al.*, 1988; Feldman *et al.*, 1988]. The implementation strained the resources of the simulator, requiring the development of a number of novel ways of using it. Some of the techniques have found application elsewhere, particularly in the related work of Goddard [Goddard], and may be of interest to other connectionist modellers. We will briefly describe them here; this section may however be skipped with no loss of continuity.

The Rochester Connectionist Simulator was designed specifically to support modelling in the structured style of [Feldman and Ballard, 1982]. As such it is optimized for highly irregular networks with many different types of activation functions and (typically) low connectivity. It is relatively inefficient for large regular networks like that described here. In particular, it maintains complex data structures describing each unit and link. Each connection requires the simulation engine to follow a long chain of pointers and make several indirect function calls. The result of this representation approach is that the original implementation occupied over 30 megabytes of memory and was unacceptably slow.

Fortunately the designers of the simulator made extensive provisions for interactions between the simulator and user code. This made it possible to modify the simulator's behavior so as to avoid paying the price for functionality that was not actually needed. The two main strategies used were the following:

virtual units All of the units in the network are indexed by (x, y) location (and perhaps other things). However, the activation rules are such that they cannot become active unless a stimulus is present at their location and/or the local clock is running. There is therefor no reason to allocate any memory for those units unless one of those conditions holds. The network implementation uses a function StimOn(x, y, properties) to turn on a stimulus at some location. As a side effect, the function

checks to see if the associated units exist and creates them if necessary. The main simulation routine notices when a clock expires and deletes the associated storage.

procedural links In the network each location has 160 Start and End units, each of which looks at every stimulus present, clock, and feedback unit in its receptive field. An implementation using the simulator's normal link primitives would therefor have been very expensive in both time and memory. The simulation avoids this cost by bypassing the normal link mechanism. The method relies on the fact that both the stimuli and the trajectories are indexed by location. In order to compute the site function for any given connection, the simulator needs to know the static weight w_{xk}, the ideal asynchrony T_{opt}, and the feedback role unit to be used to compute g_{xk}. For any trajectory unit with known direction, curvature and speed, the appropriate role unit is known and w_{xk} and T_{opt} depend only on the *relative* positions of the trajectory unit and the remote location. During network initialization the startup code constructs a table, indexed by direction, curvature and speed, that contains lists of link descriptors. Each descriptor specifies w_{xk}, T_{opt}, and the relative position of the remote location. At run time, the unit function for a trajectory unit with parameters (X, Y, D, C, S) first looks up the appropriate list using D, C and S as indices. For each entry in the list it adds the indicated positional offset to X and Y to obtain the location of the remote unit, then looks up that unit's clock, feedback and stimulus present outputs in the simulator's global data structure. It then computes the activity due to that connection as described above.

The techniques described above reduced the network size to about 5 megabytes, varying with the complexity of the stimulus. The precise running time per step has not been measured but appears to be dominated by the time required to generate the graphic display of the results.

A final extension to the simulator was made to simplify working with temporally extended stimuli. This is a facility that allows the simulator to read in a script file containing commands to be executed before specified simulation steps. This makes it possible to prepare static descriptions of apparent motion stimuli. After loading such a description the operator can single step through the simulation, letting the script take care of turning stimuli on and off at appropriate times.

The virtual unit technique described above bears an obvious relation to the cached Hough Transform techniques of Brown [Brown, 1983a] and Quiroz (described in [Ballard, 1986b]). The procedural link strategy resembles the standard way of implementing complex Hough transforms in procedural code. It is a tribute to the skill of the simulator designers that it was so easy to incorporate these mechanisms into the basic simulator without sacrificing its excellent user interface and other desirable features.

7.3 Network Behavior

The network described above interprets a large number of apparent motion displays in ways that agree with reported human phenomenology. In this section we will look at a number of examples that demonstrate various aspects of the network's behavior.

Discussing large numbers of motion stimuli in a written document presents something of a technical problem for the writer. One of the best ways to understand the network is to follow the activity of candidate trajectories as they relax toward a solution, but this requires presenting a lot of data. The examples will be discussed using a set of diagrams intended to convey as much as possible of the dynamic behavior of the system. The graphic conventions will be explained in detail in the first example, and assumed thereafter.

Two-Dot Stimulus

The simplest apparent motion stimulus one can imagine consists of two identical dots presented at different locations and times. Figure 7.6 is a representation of the stimulus, with the left side showing the physical arrangement of stimuli and the right side the temporal arrangement. Figure 7.7 shows the behavior of the network given that stimulus. Each row of the figure shows the status of some group of units at various times during the simulation, as indicated by the scale shown at the top. All of the images were generated by the simulator during actual runs.

The top row in figure 7.7 is a graphic representation of the input. The second shows the array of clock units for the network. In rows two and three each unit is represented by a square whose size corresponds to the unit's activation level. Units whose activity is zero are represented by small dotted squares. The clock units come on with the stimulus present units and decay linearly thereafter, as described in section 7.2.1. The third row shows a set of units which compute the total feature binding feedback at each stimulus location. Recall that this is the sum of four separate feedback units corresponding to the four roles that a stimulus can play. This sum of feedback units is not actually used by the network – feedback activity is presented this way to keep the display from becoming unmanageably large.

The bottom two rows show the outputs of the trajectory start and trajectory end units. Because the trajectory units have more than two parameters and code for trajectories in a non-intuitive way, the unit activity representation used for the clock and feedback units is not very useful. Instead, the trajectory unit functions draw arrows whose location and curvature are appropriate for the trajectories they represent, representing their level of activity by the length of the arrow. Start units all diverge from their shared location, while End units converge.

Let us now look at how the network's interpretation of the stimulus evolves through time. For time steps 1-15 no motion interpretation is possible, since there is

only one clock unit active in the network. At time 16 (column two in the figure) the second dot comes on, triggering the associated clock. Start units at the first location and End units at the second now begin to receive activation through their temporal site functions[2]. There are many circular arcs which are consistent with the stimulus, so initially (column 2, bottom) the trajectory space representation is highly ambiguous. Notice that the straight trajectory is noticeably stronger than its rivals, however – this is due to the curvature and length penalties in the static weight. At the next simulation step this information becomes available via the feedback units (row three, column three), and the best competitor begins to suppress its rivals. By time step 20 the curved trajectories have been driven to zero and the network is in a stable state.

Continuous Motion

One of the main points of Chapter Four was that apparent motion perception is not a specialized sense or an adaptation to occlusion, but rather represents the normal response of the motion system to an extremely impoverished stimulus. The claim is that intermediate level motion processing operates all the time in parallel with the low-level retinal motion system. It is usually not noticed, however, because its interpretations of continuous motion stimuli are heavily based on the low level system's output. Let us see how the model behaves when given a continuous motion display *without* the low level information that would normally accompany it.

Figures 7.8 and 7.9 show a very simple continuous motion input and its interpretation by the system. The stimulus sequence was made by taking the two-dot sequence analyzed above and interpolating in space and time. The second dot of the sequence (column two) produces very little ambiguity, because the dots are close enough together to fully constrain the local tangent direction. It is interesting to compare the network at time 16 (column 4) with the same time (column 2) in figure 7.7. The stimulus conditions in both cases are identical, but the interpretation in the two-dot case is highly ambiguous while in the continuous case it is much less so. This happens for several reasons. By the time the last stimulus comes on, the start unit at location 1 is well established and is suppressing all other start units at its location. It is also providing feedback to all locations in its receptive field, including the one where the new stimulus has appeared, so no start units at any intermediate location will be able to use it. At the other end, the end unit at each new location receives activity from all of its predecessors. Since the last stimulus has more predecessors than any other, it is stronger and eventually dominates. In addition, all previous stimuli are already subject to competition by End units at their locations. This keeps the relatively weak curved-trajectory End units at the new stimulus location from ever being activated.

[2]For this stimulus the asynchrony between dots is in within the optimal range for both speed units of all feasible trajectories. Thus speed is not a factor in this computation.

A simpler if not perfectly accurate way to understand the difference between what happens in the two-dot and continuous displays is to think of the feedback units as having a crude predictive value, in the sense that the network's interpretation of part of a continuous stimulus reduces the ambiguity of subsequent stimuli.

Motion Along a Curved Path

Figure 7.11 shows the network's response to a stimulus similar to the previous one except that the dots are arranged in a roughly circular pattern. As in the previous example, the Start units arrive at a unique stable interpretation very quickly. The End units, on the other hand, remain unsettled until well after the inputs have stopped arriving. This temporal asymmetry is a consequence of the way trajectories are represented. The Start unit for the desired curved path becomes dominant by time step 10 or so, and all of the stimuli appearing after that time are consistent with it. Therefor the Start representation does not need to change. On the other hand, each new stimulus is a new candidate for the trajectory end, so the End units are in an unsettled state for much of the stimulus duration. We will return to this phenomenon later, when we discuss the network's weaknesses.

Correspondence Ambiguity

Figure 7.13 shows the response of the network to a variant of the Ramachandran 'semaphore' stimulus. The usual square shape has been stretched into a rectangle. Nearest neighbor preference induces human observers to choose the vertical motion interpretation over the horizontal one. The stimulus is ambiguous both in terms of correspondence and trajectory; in the network, therefor, initial presentation of the stimulus produces diffuse activity in many trajectory units. Since straight trajectories and near neighbors have higher static weights, however, the network settles quickly into the correct interpretation.

Effect of Conjunctive Connections

As explained above, the temporally sensitive sites on the trajectory units are a type of conjunctive connection. To reflect this the network uses the product of the dynamic feature binding weights for local and remote locations to scale the input from the remote location. Figure 7.15 shows the effect of this on a display called a lambda stimulus. Here proximity is used to bias the system toward motion down and to the left. Without conjunctive feature binding, the system would settle to a state in which dot A corresponds with dot B, but End units at dot C had non-zero activation. With it, A and B are placed in correspondence and dot C receives no interpretation.

This interpretation is not in in accordance with all reports of human perception. A number of researchers report many cases in which dots appear to split [Kolers, 1972;

Ullman, 1979]. The network as described above cannot construct such interpretations. There are a number of ways one might give it the ability to do so. The simplest would be to make the conjunctive connections 'leaky', for example by using a linear combination of the sum and the product of the dynamic weights. A more sophisticated approach would be to let the correspondence failure trigger a higher level explanation-seeking process, which might construct explanations involving splitting or occlusion. This approach would be consistent with a general opinion of the author's, that such events are common in perception – that low-level processes handle as much of the input as they can, appealing to higher level disambiguating processes for help when they detect a convergence failure.

Effect of Negative ISI

When the ISI for a dot pair is negative (*i.e.* the second dot appears before the first has disappeared) observers report that the impression of motion is weakened. In the network this is reflected by the fact that a given location cannot be used in its role as a trajectory start until the stimulus present unit has gone off. Figure 7.17 shows the effect of this on another variant of the semaphore stimulus. Here the offsets of the first two dots and onsets of the second two are staggered in time. The result is that the network chooses an interpretation in which the dots move along the long side of the rectangle rather than the short side as they would normally do. This happens despite the fact that the activity of the clock units is almost identical to that in figure 7.13, above. The reason it occurs is that the negative ISI temporarily keeps dot A from corresponding with dot B. This gives dot C a head start in the competition. When dot A is free to compete, it is unable to overcome C's lead, so it corresponds with D instead.

Simple Biases

One of the earliest observations about apparent motion stimuli was that observers have a great deal of conscious control over their interpretations [Wertheimer, 1912]. In ambiguous cases they can flip back and forth between interpretations, just as they might do with a Necker cube. This sort of influence is modelled in the network via the bias inputs to the trajectory units. Figure 7.18 shows the effect of applying a bias of 2 to a pair of curved trajectory units for the two-dot stimulus of our first example. As stated in Chapter Six, bias inputs effectively change the input weights of a given unit, in this case making the curved trajectory units more sensitive to their input. The result is that the curved trajectory suppresses the normally stronger straight trajectory.

The same mechanism could obviously be used to incorporate slip information from the low level motion system. The information would be abstracted by the segmentation system into some low-parameter description. It would be made available

during the 100 to 200 millisecond integration time of the low level system, which (as in the negative-ISI example above) would allow it have a strong impact on the eventual solution found by the network.

7.3.1 Making Use of Stimulus Properties

As we have seen, the network described above can handle a variety of basic continuous and apparent motion stimuli. Additional mechanism will be needed to handle more complex aspects of the phenomena. This section will describe the mechanisms needed to handle property information.

Categorical Properties

We will look first at properties which can be compared by simple metrics that return only an indication of match strength. The paradigmatic example is color. Consider a Ramachandran semaphore stimulus in which color is the only cue as to which interpretation is correct. How might that information be used? The naive solution would be to replicate the entire network for each property value, effectively parameterizing the clock and trajectory units by color as well as their other characteristics. (This would be an example of value unit representation.) Because red blobs *can* correspond with blue blobs, the red trajectory units would have to be connected to blue clock units with some reduced weight, et cetera. This process would have to be repeated for every categorical property recognized by the system. It seems clear that this solution would require an impractical number of units.

The alternative to a pure value unit approach is to compute property match information separately and use it to influence the relaxation of the trajectory units. This leads to a classic problem in connectionist representation. There are not enough units to represent the match quality between all possible pairs of location in the image. The network must give up specificity of the representation along some dimension. This in turn will produce crosstalk problems – that is, the network may be able to tell that the scene contains matching locations but have no representation of which ones they are. This is another example of the connectionist counting problem, and the techniques of Ballard and others remain relevant. The approach here will be to use attention to restrict the comparison process.

The network will be augmented with a single buffer that can store a property vector like those associated with the input locations. When a stimulus is presented, one of the blobs will be selected and copied into the buffer[3]. Each location will compare itself to the buffer and compute a match quality. The output of the match quality unit will be used to make its location more salient using the feature biasing

[3]The process responsible for selection can be thought of as a sequential but preattentive process, along the lines of the visual routines of Ullman [1985].

mechanism described in chapter Six – that is, by modifying the local feedback activity in such a way as to effectively increase the static weights from that location to all trajectories. Figure 7.19 shows the general shape of the mechanism.

Figure 7.20 shows the effect of the mechanism on a semaphore stimulus in which the upper and lower stimulus pairs have different properties (represented here by open and closed squares.) Property information thus favors a horizontal interpretation, contradicting the system's normal tendency to prefer nearest neighbors. At time 1 the upper left stimulus is selected and copied to the global buffer. All locations compare themselves to the buffer, including the one from which the stimulus properties originally came. Thus at time 16 there are two matches, at upper left and upper right. This strengthens both the horizontal and the vertical interpretations. However, the vertical trajectory units see only one enhanced stimulus, while the horizontal units see two. This permits the upper horizontal trajectory to dominate its vertical rivals, which in turn clears the way for the lower horizontal trajectory to emerge.

This notion of how property match should work raises a number of issues with regard to the use of attention in visual matching. In particular, it may conflict with Triesman's well-known results on the difficulty of search for conjunctive features [Triesman, 1985]. If the visual system had a mechanism for comparing arbitrary feature vectors in parallel against all locations in the visual field, one would expect search for conjunctions to be parallel. The property match mechanism described above can accommodate Triesman if the match mechanism is defined more narrowly. Think of the property vectors associated with the input locations as forming a stack of planes, each coding for a particular value of a particular property. Suppose that the match process consists of selecting not only a particular stimulus, but a property of that stimulus as well. For example, a visual routine seeking the fate of some stimulus might notice that *redness* was one of its salient properties, and instruct all feedback units to use the *red* property plane as a bias.

It should be noted in connection with property matching that the Triesman paradigm continues to produce surprising results, confounding the clean picture that seemed to be emerging in her earlier work. Some surprisingly complex phenomena have turned out to be preattentively segregatable, including such things as shape-from-shading convexity [Ramachandran, 1988]. In addition, it appears that conjunctive search is parallel provided the right properties are conjoined. In particular, shape/motion, color/motion and convexity/color can all be searched for in parallel [McLeod *et al.*, 1988; Tiana *et al.*, 1989]. These results suggest that there is a lot left to be learned about visual matching and conjunctive search.

Shape Properties

We will now turn to the more complex problem of how form information can be used to influence the selection of trajectories. The paradigmatic example of the effect

is Foster's stimulus, which demonstrates that the orientation of stimuli in apparent motion strongly affects the path over which the stimuli appear to travel. Other examples include the results of Shepard, Attneave and others on rigid vs. non-rigid tradeoffs in apparent motion.

The general approach will be based on the transformation networks of Ballard [1984]. Figure 7.21 shows the structure and connections of the proposed solution. As before, the network will use a global buffer to store the property information associated with a selected stimulus. In addition the buffer will store an encoding of the selected stimulus position. (This is not at all burdensome - such an encoding would be needed anyway to specify which location is selected.) The transformation network will compare the shape descriptors at each location to the global buffer and determine the positional offset $(\Delta x, \Delta y)$ and orientation difference $\Delta\theta$ between them[4]. Simple geometry shows that those two pieces of information specify the radius of the circular trajectory that maps one stimulus onto the other. This radius will be used to bias all trajectories with consistent radii. Both the transformations and the radius estimates will be expressed as confidences, *i.e.* in a value unit representation with activity encoding the strength of the evidence. This will allow the network to handle objects with multiple axes of symmetry.

In order to perform the computation described above the network must be able to compute the relative orientation of two stimulus blobs. That in turn means that it must have a way of representing the shapes of the blobs. The representation used here is intended only to help disambiguate trajectories, and not as a model of how humans represent shape. A more realistic and general shape representation would almost certainly support the same types of computations, however. What the motion system requires is a compact shape descriptor that makes it easy to determine the extent to which one shape resembles another under rotation through various angles. This is related to the problem of determining axes of symmetry of a figure; in the latter problem one attempts to determine to what extent a figure resembles itself under reflection about various axes. Friedberg [1984] studied that problem empirically in the course of work on recovering surface orientation from skewed symmetry. The shape representation used here is based on the best of his algorithms, which he called the *sector symmetry* evaluator. Like most of his methods, the sector symmetry evaluator measures necessary but insufficient conditions for true symmetry. Thus it may report a similarity where none is present, but it never reports dissimilarity when two figures are in fact identical.

The sector symmetry evaluator represents shapes as vectors of length n. The

[4]The $(\Delta x, \Delta y, \Delta\theta)$ network can be relatively simple because the problem it solves is fundamentally easier than that addressed by the general form of Ballard's transformation networks. This is because of the assumption that the shape descriptors are always referred to the centroids of the blobs. The result is that distance and orientation difference are completely independent. If the shape descriptors did not fully specify the location of the blob, the translational and rotational components of the transform would interact and the network's job would be much harder.

vector is computed by finding the shape's centroid and drawing lines through it, splitting the shape into n sectors (pie slices) subtending $2\pi/n$ radians each. The mass of the ith sector becomes the ith entry in the vector. Let A and B be the vectors describing two shapes. To determine the extent to which B looks like a rotation of A, one computes what will be referred to as the *sector similarity* evaluator

$$E_{ss}(\phi) = \sum_{i=1}^{n} \left| A_i - B_{(i+\phi) \bmod n} \right|$$

for all values of ϕ between one and n. Clearly if A and B are identical under rotation, E_{ss} will be minimized by the value of ϕ that relates them (ignoring errors due to undersampling).

The transformation network as implemented consists of an array of units quantized by Δx, Δy and $\Delta\theta$. The Δx and Δy units encode integer values on the interval $[-7, -7]$. The shape vector divides the shape into sixteen sectors, giving an angular resolution of $\pi/8$ for $\Delta\theta$. Each transformation unit compares the blob descriptor in the global buffer to the descriptor at the appropriate offset, and computes a match strength:

$$M(\Delta\theta) = \frac{\sum_{i=0}^{n}(A_i + B_i) - E_{ss}(\Delta\theta)}{\sum_{i=0}^{n}(A_i + B_i)}$$

Since E_{ss} cannot be less than zero or greater than the sum of the masses of the two blobs, $M(\Delta\theta)$ is always on the interval $[0,1]$.

Any given transformation unit corresponds to rotation along a path with some radius. For each of the five values into which trajectory radius is quantized, the network computes the the maximum match strength over all of the corresponding transformation units. That maximum value is used to bias all trajectories with the appropriate radius. Since the transformation unit activities are all between zero and one, the effect is to weaken trajectories whose radii are not consistent with any trajectory involving the selected stimulus.

Figure 7.22 shows the effect of shape information on the interpretation of a highly asymmetric stimulus. The transform network finds strong evidence for rotation (clockwise) through $\pi/2$ radians, weak evidence for rotation through $-\pi/2$ radians, and essentially no support for any other rotation. Figure 7.23 shows a plot of match strength versus angle for the stimulus of Figure 7.22.

Figures 7.24 and 7.25 illustrate a rigid/non-rigid tradeoff like those noted by Shepard. When the stimulus is presented with an SOA of sixteen time steps, the network settles on an interpretation involving rigid motion along a curved trajectory. If the SOA is shortened to ten time steps, it chooses a shorter straight trajectory that implies some shape change. This occurs because at the shorter SOA the long curved trajectory implies a speed that is too high for the optimal range (*i.e.* that does not fall between the canonical speeds for the two speed units). Since the stimulus has a fair amount of self-similarity at a rotation of zero, the bias from the shape system

is not great enough to compensate for the longer trajectory's weakness. Thus the shorter trajectory wins the competition.

As usual, the network could be modified in a number of ways to trade off resolution and parallelism against size and complexity. Any of the standard methods could be used to reduce the size of the $\Delta x/\Delta\theta$ parameter space. If it proved necessary for the bias to be directed to particular candidate trajectories rather than broadcast to the whole trajectory space, one could apply it through a conjunctive connection that looks at the selected stimulus location and Δx. This would direct the right bias to the right locations. Another way to achieve the same effect would be to allow attention to gate the connections between input stimuli and the transformation network.

A more interesting modification would be to generalize the $\Delta\theta$ computation to include more general shape change, particularly including size and some parameterization of distortion. Size change could be used to bias trajectories in depth. It would then be a small step to make the connection between shape change descriptors and trajectory units bidirectional. The network would then be able to work backward from the winning trajectory to recover the shape change undergone by the segment. That is, the network would start out with ambiguous estimates of both trajectory and shape change, and would relax to unambiguous and mutually consistent estimates of both.

7.3.2 Advanced Topics

This concludes discussion of the set of effects that can be handled by the existing implementation. This section will suggest ways in which a variety of other influences could be incorporated. The mechanisms needed to incorporate these influences for the most part involve interactions with parts of the visual system that have not been built, such as high level motion perception and the segmentation process. It is for this reason that they have not been included in the present implementation.

Blocking and Occlusion Static objects in a scene interfere with trajectories through their location [Kolers, 1972]. The usual result is that stimuli are seen to move around the blocking object in a curved path or to pass in front of or behind it. A simple way to incorporate this into the network would be to let static objects provide a small amount of negative bias to trajectories whose receptive fields overlap their locations. The amount of negative bias should decrease with distance, like the static weights. If the scaling is done carefully, blockers lying between two stimuli will inhibit them enough to make other, perhaps curved trajectories more plausible. This would however require fairly precise control of the inhibitory bias. An alternative method would avoid this problem by imposing an order on input sites. Occluders would apply a signal that prevented activation from flowing from from remote locations on one side of

them to trajectory units on the other. This idea is similar to the shunting inhibition mechanism proposed by [Torre and Poggio, 1978].

There are a number of reports of stimuli which appear to move behind static objects when they disappear [Ramachandran *et al.*, 1986]. Usually the display makes use of some strong cue, such as entrainment or having the stimulus reappear on the other side of the occluder. The interpretations can be quite clever; for example, the system apparently takes into account the shapes of the stimulus and static object in deciding whether occlusion is a plausible interpretation. In the model a trajectory cannot be activated without active clock units at both ends of the trajectory. In order to handle occlusion, therefor, the network would require a process that turns on a 'virtual stimulus' at the location of the occluder. In view of the sophisticated shape and visibility computations that seem to be involved, one would expect the process to take place at a fairly high level. It is difficult to say more about it without some notion of how shapes are represented.

Shepard Paths Chapter Two mentioned the results of Shepard and Zare [1982] on the use of low-contrast paths as a cue to the trajectory followed by an apparent motion stimulus. Several different mechanisms are probably involved in producing the effect. Some are very low level, and could be fitted into the current model easily enough. Clearly the flash of the low contrast path will have some effect on the low-level motion system. Strictly speaking, the path should not stimulate the low level system's directionally selective units, since it has no spatiotemporally oriented energy. One would however expect it to provoke at least some response from the low level, raising its response above background. In addition, it would not be surprising to find that the low level system detects flashes and can interpret them as motions whose speed falls outside the operating range of the system. This idea is supported by results of Kelly and Burbeck [1987], who used an adaptation paradigm to show that at very low spatial and high temporal frequencies the human motion detection system is insensitive to direction. In short, it is likely that part of the effect of the paths comes from the normal facilitory effect of the low level system. This cannot be the whole story, however. Madden's experiments showed that the Shepard and Zare effect is quite intolerant of variations that a low level system could not be expected to notice. In particular, the shape of the path must be consistent with the shape of the stimulus. Also, the path contrast must be consistent with the motion interpretation; if it is reversed, or is of the correct sign but too high, one gets the impression of motion along a straight path while an unrelated object flashes on the screen [Madden, 1989a].

As in the case of occlusion, the conditions under which motion is seen seem too strict for any mechanism but a relatively high level hypothesize-and-test process. Interpretation might proceed like this: a dot/flashed-path/dot sequence stimulates the same set of trajectories as a two-dot stimulus. The flashed path is not interpreted as a correspondence candidate because it is on for only a very short time. It does however stimulate the low-level system to some extent, making the trajectory defined

by the path at least somewhat plausible. The sequence also stimulates the highest level, activating a scenario for fast motions. Finally, an attentive process is invoked to perform the shape analysis and decide how to classify the event. The interpretation of the path (either 'fast motion' or 'extraneous event') is fed back to the trajectory representation, biasing its final decision as to the dot trajectory.

Facilitation Over Time It has long been known that the strength of apparent motion percepts is greatly enhanced by repetition. Wertheimer [1912] noted that after several repetitions of a two-dot stimulus, subjects continued to see motion (at least for a few cycles) even if the second dot was omitted from the display. The implementation described above does not show this effect; each presentation of a stimulus is interpreted *de novo* and results in the same relaxation process. One would like the network to learn what to expect from repeated stimuli, so that it could fill in for missing data and converge more quickly on later presentations of the stimulus. The obvious way that this might happen in the model is via interaction with the highest level of the motion system. Suppose that the highest level takes approximately the form described by Goddard [Goddard]. That is, the fundamental process underlying scenario recognition is matching of events in the scene against finite-automaton-like descriptions of scenarios stored in long-term memory. Assume also that the high level also includes a short-term memory for scenarios – Goddard's model would presumably need some such mechanism to support learning. The first few presentations of a stimulus would produce a short-term scenario description; after that, top-down feedback via the mechanisms described in Chapter Six would cause the system to converge quickly.

Field Effects and Entrainment Ramachandran and Anstis have done a number of interesting experiments with displays containing multiple copies of an ambiguous apparent motion stimulus [Ramachandran and Anstis, 1983b; Anstis and Ramachandran, 1986]. The general finding is that in such situations the visual system has a strong tendency to assign the same interpretation to all of the stimuli in a scene. In addition, when observers use attention to force one of the stimuli to flip from one interpretation to another, all the other stimuli in the field flip at the same time. In terms of the model this can be interpreted as another instance of the action of the highest level of the motion system. In Goddard's model the scenarios that describe events are not spatially indexed. A layer of binder units (see [Feldman, 1982]) is used to form temporary associations between scenarios and spatially indexed stimuli. If the high level can only have one scenario engine active at a time, the field effects arise trivially. Attending to one stimulus in the field activates a scenario describing it. Since the other stimuli are also consistent with the scenario, however, their binders become active as well. They therefor receive the same top-down input and are assigned the same interpretation as the attended stimulus.

This view of the causes of entrainment predicts that relative phase of the individual stimuli should matter. Consider a field consisting of a number of Ramachandran rectangles like that shown in Figure 7.12. If they alternate synchronously, they will be given a common interpretation as described above If however they all move at random phases and frequencies, they will no longer match the high-level scenario and the entrainment effect should be weakened.

7.3.3 Failure Modes

One of the hazards of implementing one's theories is that one is forced in a particularly direct way to acknowledge their shortcomings. The network described in this chapter exhibits a number of different types of failures. Some of these are infelicities of implementation. This class includes such things as errors due to the coarse quantization of trajectory space. The solutions to problems of this type are generally obvious, so they will not be discussed here. The interesting failures are those that are due to problems with the network design. For example, it turns out that the roles represented by the various types of feedback units aren't quite right. These design problems come in varying degrees of seriousness, though all of them appear to be solvable.

Stimulus Roles Any feature binding network can be thought of as seeking explanations for its features. That is, it settles toward a state in which every unit is 'cancelled' by an active feedback unit. In the network described in this chapter feedback is split into four roles corresponding to the four types of connection each location can make with a trajectory unit. Unfortunately the set of roles used here does not span the space of *logical* roles. As stated in section 7.2.2, locations in the middle of a continuous trajectory have neither local Start nor local End units active, though they are receiving feedback in their remote Start and remote End roles. This leaves them free to compete for local Start or End roles in other trajectories when new stimuli appear. The problem was handled here by allowing remote units to weaken their local counterparts, but this is not an ideal solution. It would be more correct to recognize 'intermediacy' as a role in its own right, one that is inconsistent with being a trajectory start or end. This would require some cleverness, since a location that is an End at one instant should be easily convertible into an Intermediate but not vice versa. Done right, however, it would solve the messiness of the End unit competition that was noted in the curved continuous trajectory example.

Specifically, each location could be given a single 'intermediate' feedback unit, which would compute the OR-of-AND of consistent pairs of remote Start and End units. Thus it would fire only when there were good explanations both for where the local stimulus came from and for where it went. It would inhibit local Start and End units by adding its output to their respective feedback role units. The result would be

better behavior in several situations. First, intermediate locations along a trajectory would no longer respond to the later appearance of unrelated stimuli. This would happen without damaging the network's ability to handle changes in direction (*i.e.* the case where a given stimulus serves as the end of one trajectory and the start of another.) Second, in continuous motion the appearance of a new stimulus along an established trajectory would quickly and smoothly supplant the previous End unit, rather than engaging in a protracted struggle with it as in the current design.

Conjunctive Feedback As mentioned in the discussion of the lambda stimulus, the network is too aggressive about insisting on a unique assignment of trajectories to stimuli. It does not allow the splitting noted by Kolers and others[5]. The previous discussion suggested ways that the feedback mechanism could be altered to permit splitting, but it is not clear that this is the right level at which to handle the effect. The idea of a given stimulus splitting obviously involves the notion of identity over time, so it may well have as much to do with segmentation as with motion per se. In that case it cannot be fully understood without a theory of how motion and segmentation interact, which takes it beyond the scope of the present work.

Overenthusiastic Competition The feature binding technique was originally conceived as an extension of the Hough transform, which is designed to map one static parameter space into another. The development ignored the fact that in the motion application time is both a dimension of interest and the domain in which the relaxation takes place. This leads to two problems. The first is that although the Start/End units have temporally sensitive sites, they send feedback impartially to every stimulus within their *spatial* receptive fields. This may result in a trajectory competing for a location that it isn't actually using – one for which the age relative to its own age is inappropriate. An example of this can be seen in the negative-ISI example presented in Figure 7.17. Note that at time 15 there is feedback activity at the locations of the upper stimulus pair, due to residual activity of curved-trajectory explanations for the lower stimulus pair. Even if the upper pair had moved at time 15, therefor, the upper trajectory units would not have begun to compete until time 17, when the residual activity of the lower curved trajectories disappeared. As a general rule, feature binding networks in which the explanation units are non-linear (*e.g.* those that use conjunctive connections) require corresponding nonlinearities in the feedback units. That is, each feedback unit must simulate the associated explanation units well enough to determine how much activity they are deriving from the local feature. In the case of the motion network, this means that they should use the same

[5]In fact a perfectly balanced ambiguous stimulus will be interpreted as splitting, as noted in the line-finding example of Chapter Six. Stimuli which are nearly balanced will remain ambiguous for a long time, because the rate at which the stronger interpretation suppresses the weaker is proportional to the difference between them. Thus a certain amount of splitting will arise naturally.

sort of temporally sensitive sites that are used by the trajectory units. This would limit trajectory units to competing for those locations that have appropriate relative asynchronies.

Another problem with feature binding over time is its finality. Recall that the dynamic weight g_{xk} is the ratio of an explanation's own activity to the total activity of explanations for the stimulus in question. The only exception is that when the ratio is zero over zero, the weight is defined to be a constant. This means that an existing weak explanation can 'lock out' a potentially better explanation that depends on a newly appeared stimulus. Roughly speaking, one can say that the system has gotten stuck in a local maximum of its objective function. This could be fixed by making another exception to the feature binding rule, so that newly appeared stimuli are given special privileges. For example, one might let the activation a trajectory unit derives from remote stimuli be the sum of two components, one being the standard feature binding activation assumed above. The other would consist of the standard temporal goodness of fit multiplied by a static weight that decays rapidly after stimulus onset.

Stimulus Masking The last problem with the network design arises because a particular location in the input array can store information about only one blob at a time. When a blob appears at some location, any information about what it replaces is destroyed. For many apparent motion stimuli this does not matter, but in some cases it leads the network to make interpretations that differ sharply from what human observers report. One example is the rotating blobs of Shepard, in which changes in the orientation of an irregular polygon are interpreted as rotations around the polygon's center. There is no reason why the trajectory representation used in the current network could not be extended to rotations – they would simply become curved trajectories of zero radius. The problem is that there would be no way to activate such a trajectory. The information describing the second stimulus would wipe out the description of the first. Even relative age would be lost, because the second stimulus onset would reset the local clock.

One way to fix the masking problem for Shepard's rotating blobs would be to allow the second stimulus to be compared to a copy in the global buffer. Presumably the global buffer would be needed anyway in order to compare the orientations. This would not help for all situations, unfortunately. A classic apparent motion stimulus that the current design cannot handle is Ternus' stimulus (see Chapter Two), in which the observer must choose between rigid motion of a group of three dots and motion of one dot in a long trajectory while the other two remain stationary. In order to arrive at the group-motion interpretation it seems that the network must be able to represent the concept of one blob being replaced by another at the same location. A possible way to do this would be to loosen the notion of 'location' in the input array. Suppose that the input array contains a pool of blob descriptors at fairly high spatial resolution. When a stimulus appears, the descriptors compete based on proximity for the right to represent the associated information. The winner preserves the stimulus

description for as long as its clock unit remains active, even if the stimulus itself goes away. Thus if a second stimulus appears at the same location, some nearby descriptor will win the competition. In this view, the ISI dependence of Ternus' stimulus arises because when the ISI is very brief, the flicker of the blob may be classified as a property of the original blob rather than replacement of one blob by another. In that case no descriptor would be generated and the second blob would not be available to enter into trajectory competition.

7.4 Conclusions

We have shown that the conceptual design of Chapter Five can in fact be elaborated into a working program. Although the resulting network has its problems, it handles a broad class of intermediate-level motion stimuli quite well, and could clearly be extended to cover more with additional mechanism. In this final section we will review the set of phenomena that it can account for.

Continuous Motion The network was designed from the start to avoid the division into frames that is characteristic of many previous models of motion understanding. It does not distinguish between 'real' and apparent motion in any way, except that in the former case it makes use of low-level motion information to help the relaxation along.

Korte's Third Law Because it represents motion as trajectories at constant arc speeds, the network cannot interpret stimuli whose asynchrony is too short for the fastest speed in the parameter set. It thus exhibits the relation between minimum SOA and distance noted by Korte. Because the trajectory unit responses are a graded measure of goodness of fit, the minimum SOA is not only a function of distance; other evidence can compensate for a marginally implausible asynchrony.

U-shaped Curve By the same token, the network cannot represent trajectories whose speed is too low. The result is the so-called U-shaped curve for asynchrony. Motion is seen only within a certain range of asynchronies, outside which the stimuli will be seen as unrelated or weakly related phenomena.

Correspondence and Trajectory Rivalries In apparent motion cases the network tends to choose interpretations in which each dot in the first frame is seen to move along a unique path to a unique location in the second frame. Because the the trajectory representation is distributed and the competition mechanism is not terribly specific, complex scenes with many stimuli may receive chaotic interpretations. That

is, the network will converge to a state in which most stimuli have multiple partially active interpretations.

A Priori Trajectory Preferences When the input is ambiguous, the network tends to prefer correspondences between nearest neighbors and motion along straight trajectories.

Incorporation of Low-Level Motion Information Within the limits of the representation of segments, the network provides a natural way to allow low-level motion information to be factored into its interpretation of a scene.

Attention and Expectation Likewise, there is an obvious way to incorporate top-down information into the computation.

Tradeoffs In all cases (Korte's Third Law, nearest neighbor preferences, retinal slip, expectation) tradeoffs occur. The network's interpretations are a result of combining all available information, and strength in one cue can compensate for weakness in others.

Stimulus Properties Stimulus properties such as color can influence correspondence, but only if the observer attends to the appropriate property.

Stimulus Shape The shape of a stimulus affects what trajectory it follows to arrive at its destination.

Other Behaviors Although the implementation does not handle them, plausible mechanisms exist for adding entrainment, blockers and occluders, and Shepard paths to the set of influences that the network makes use of.

7.4.1 Summary

We have seen that the network can handle a large set of classical apparent motion stimuli. It is not perfect, and in retrospect there are parts of it that clearly should be designed somewhat differently. However, all of the problems that have been found to date have plausible solutions. So far there do not appear to be any deep conceptual problems that would invalidate the architecture or the general design. By far the least satisfactory aspect of the design is its the fact that it says nothing about how segmentation is accomplished or how it interacts with motion perception. This is unavoidable given the state of current knowledge in this area.

Certainly one could go on elaborating the model and the network design for quite some time, for example by implementing some of the fixes and extensions proposed above. However, it is not clear that the intellectual payoff would be worth the effort. The primary goals of the research have already been accomplished. In particular, it is clear that the overall design and connectionist techniques that were developed through chapters Four, Five and Six can indeed produce a highly functional motion understanding network. The next chapter will summarize the work from a broader perspective, draw conclusions and discuss future research directions.

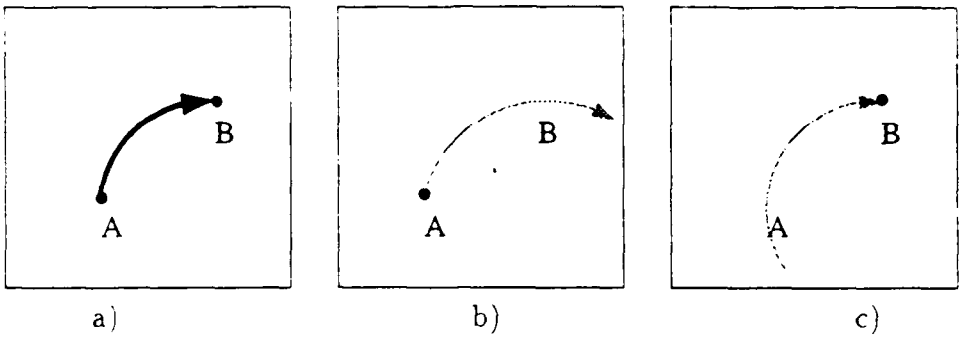

Figure 7.1: Encoding a trajectory using SE units. a) trajectory. b) Start unit. c) End unit.

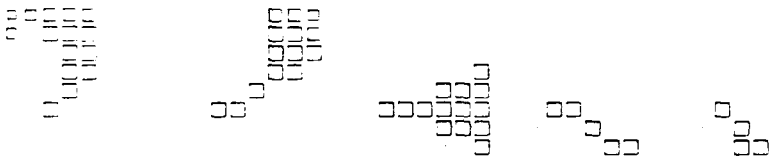

Figure 7.2: Receptive fields for SE units with various values of curvature for a fixed location and tangent direction.

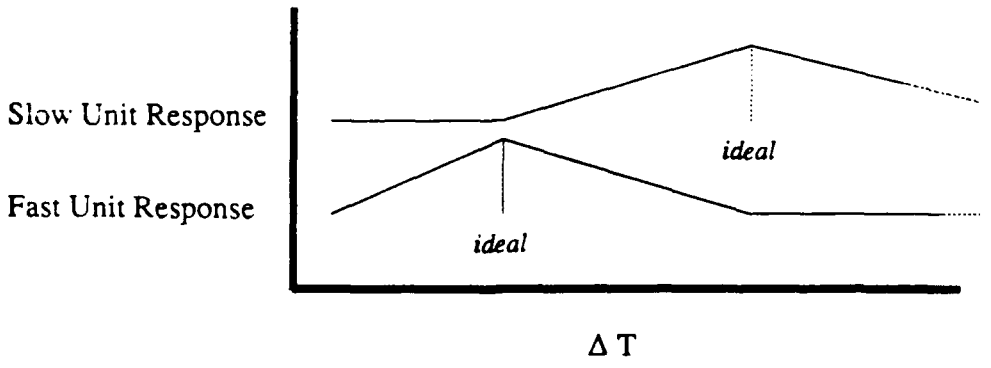

Figure 7.3: Response of canonical Slow and Fast SE unit sites as a function of measured Δt.

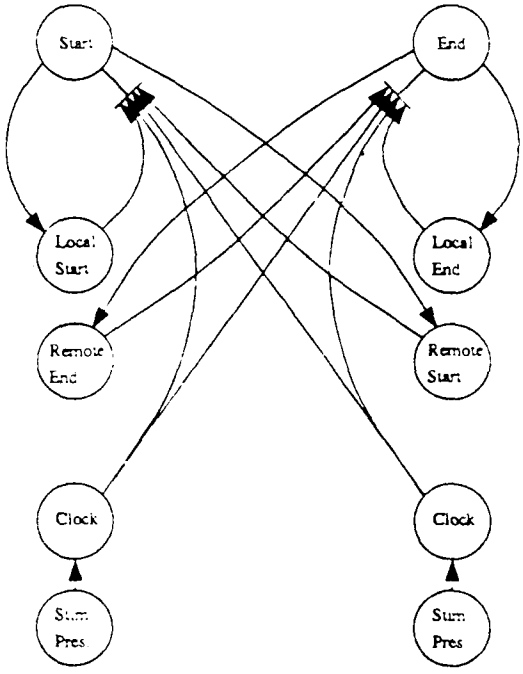

Figure 7.4: Units and connections for a single trajectory.

Start/End Unit Computation

For each site

1. Compute temporal goodness of fit
2. weight by w_{xk}
3. weight by g_{xk}

Sum over all sites
Multiply by product of biases

Figure 7.5: Summary of trajectory unit computation

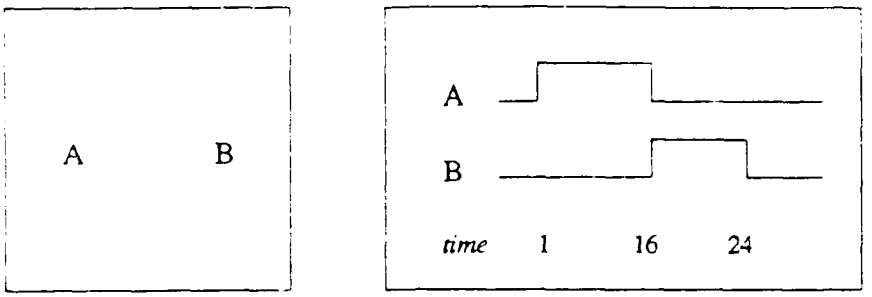

Figure 7.6: Two-dot display, spatial (left) and temporal (right) arrangement of stimuli

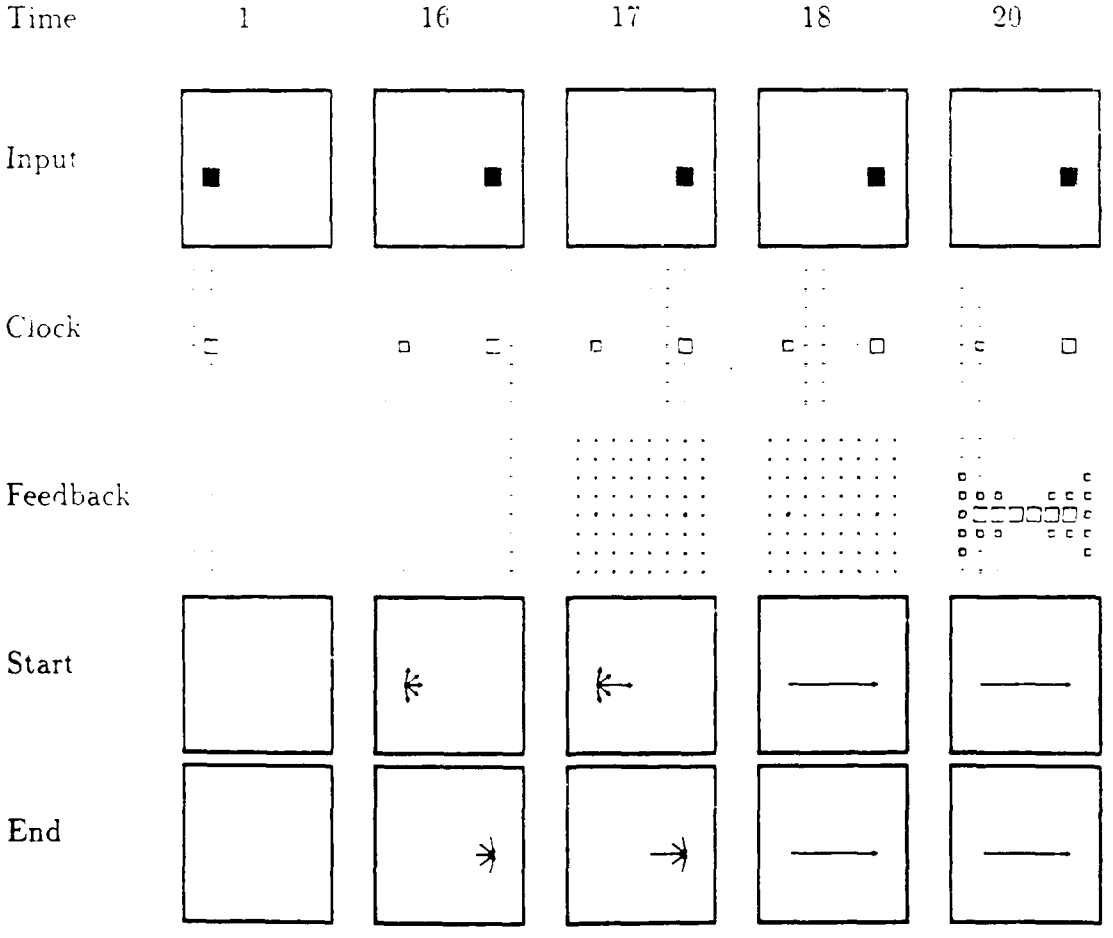

Figure 7.7: Interpretation of a two-dot stimulus

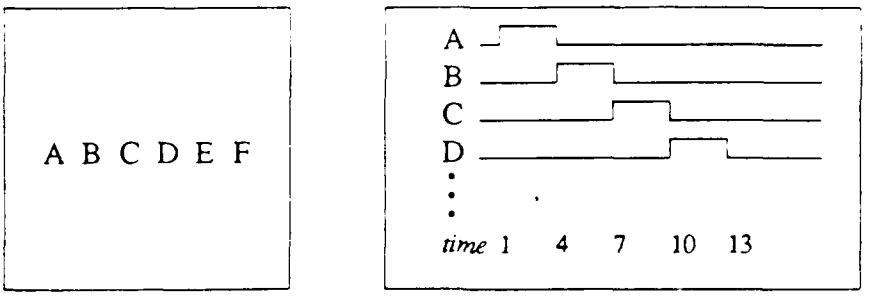

Figure 7.8: Continuous motion: spatial (left) and temporal (right) arrangement of stimuli

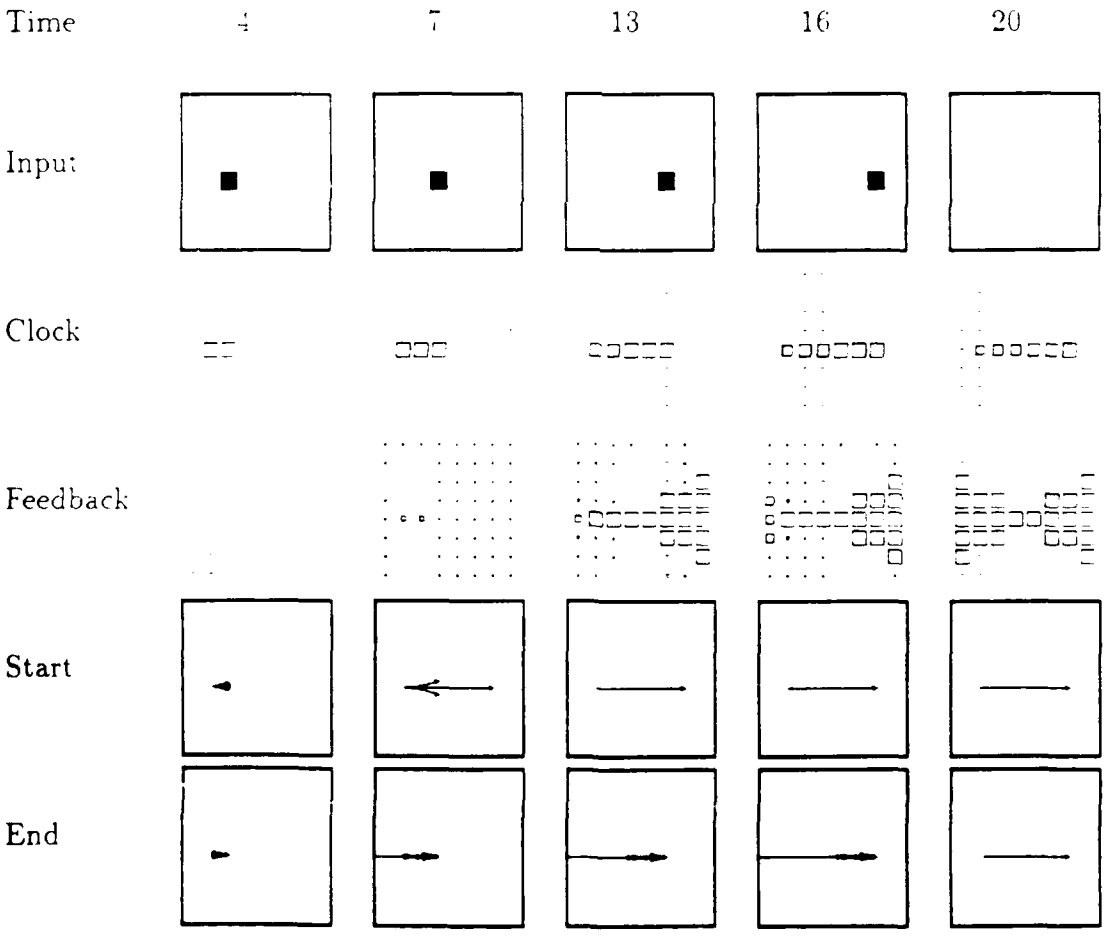

Figure 7.9: Interpretation of a continuous motion stimulus

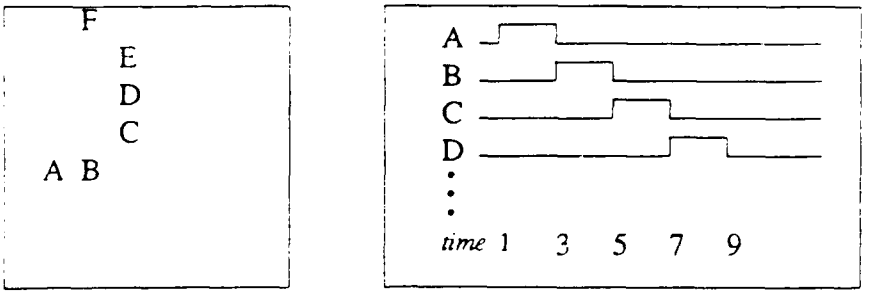

Figure 7.10: Continuous motion along a curved path: spatial (left) and temporal (right) arrangement of stimuli

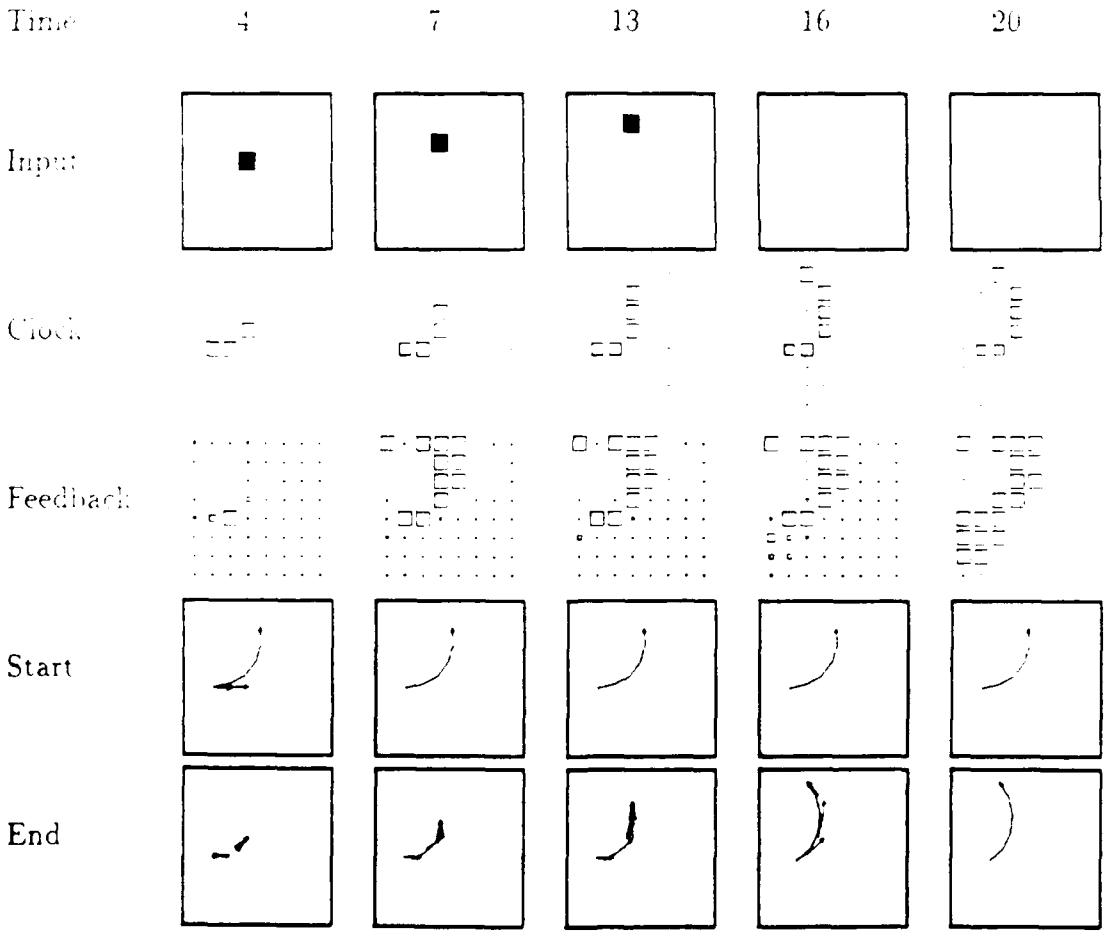

Figure 7.11: Interpretation of continuous curved path stimulus

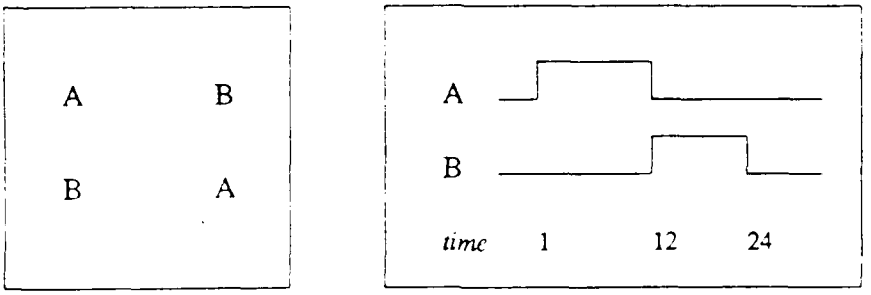

Figure 7.12: Ramachandran Semaphore: spatial (left) and temporal (right) arrangement of stimuli

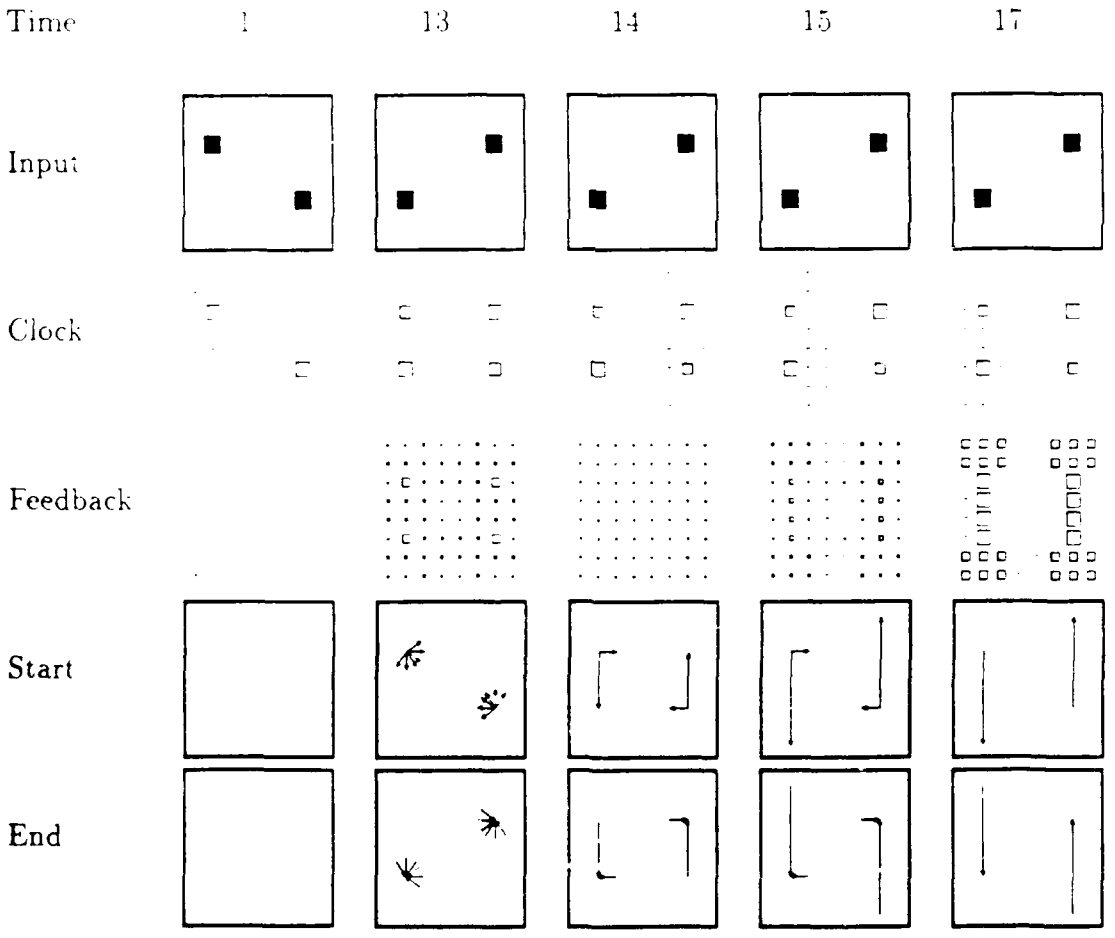

Figure 7.13: Interpretation of a Ramachandran semaphore stimulus

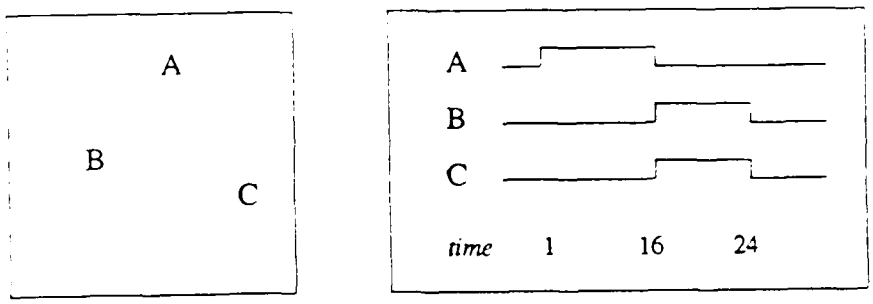

Figure 7.14: Lambda stimulus

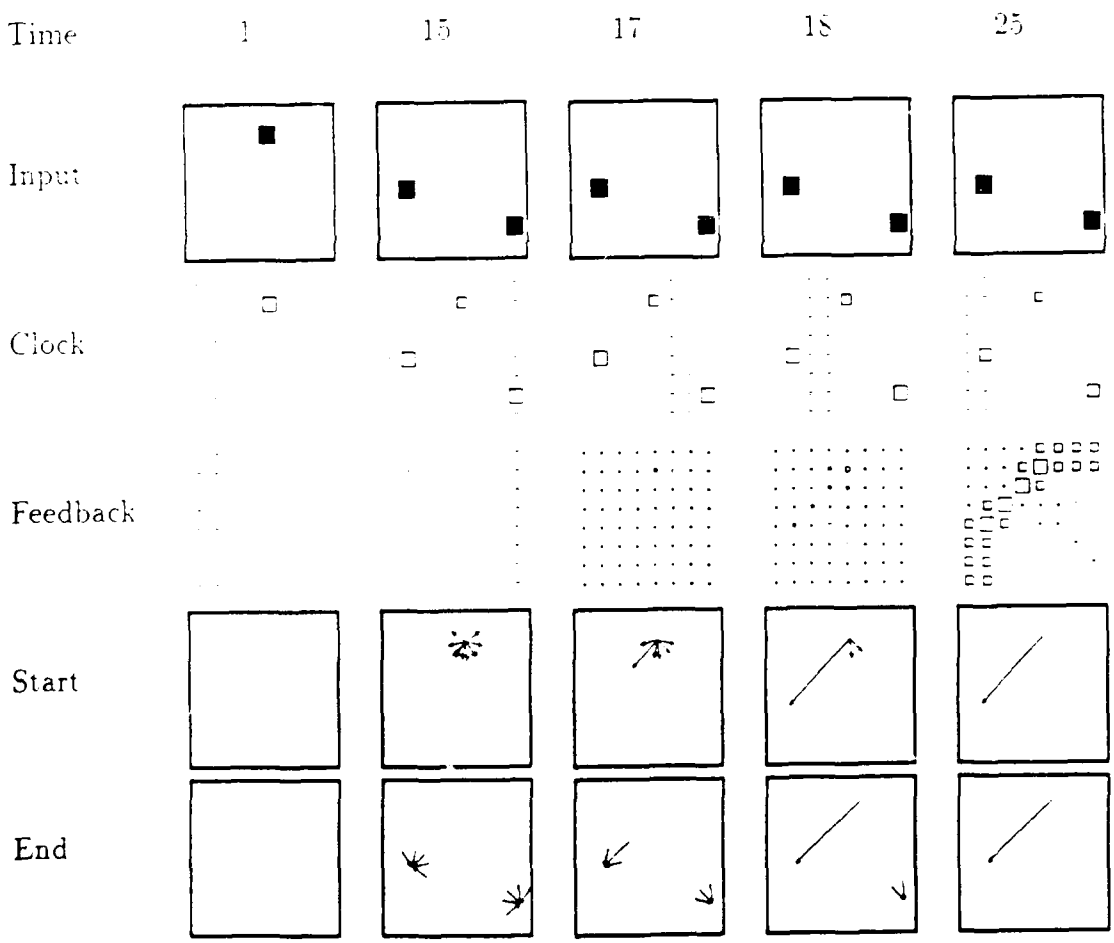

Figure 7.15: Lambda stimulus interpretation, showing the effect of conjunctive connections.

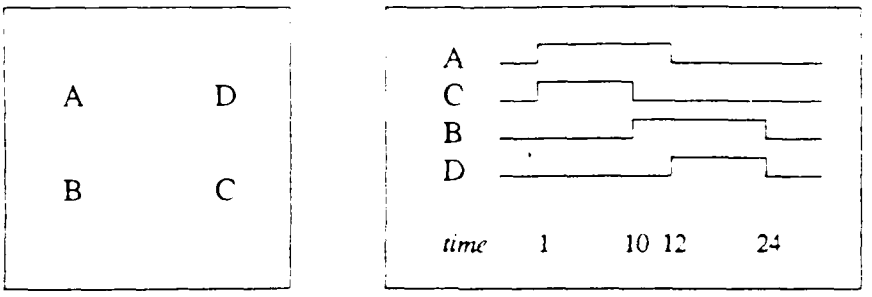

Figure 7.16: Semaphore stimulus with negative ISI

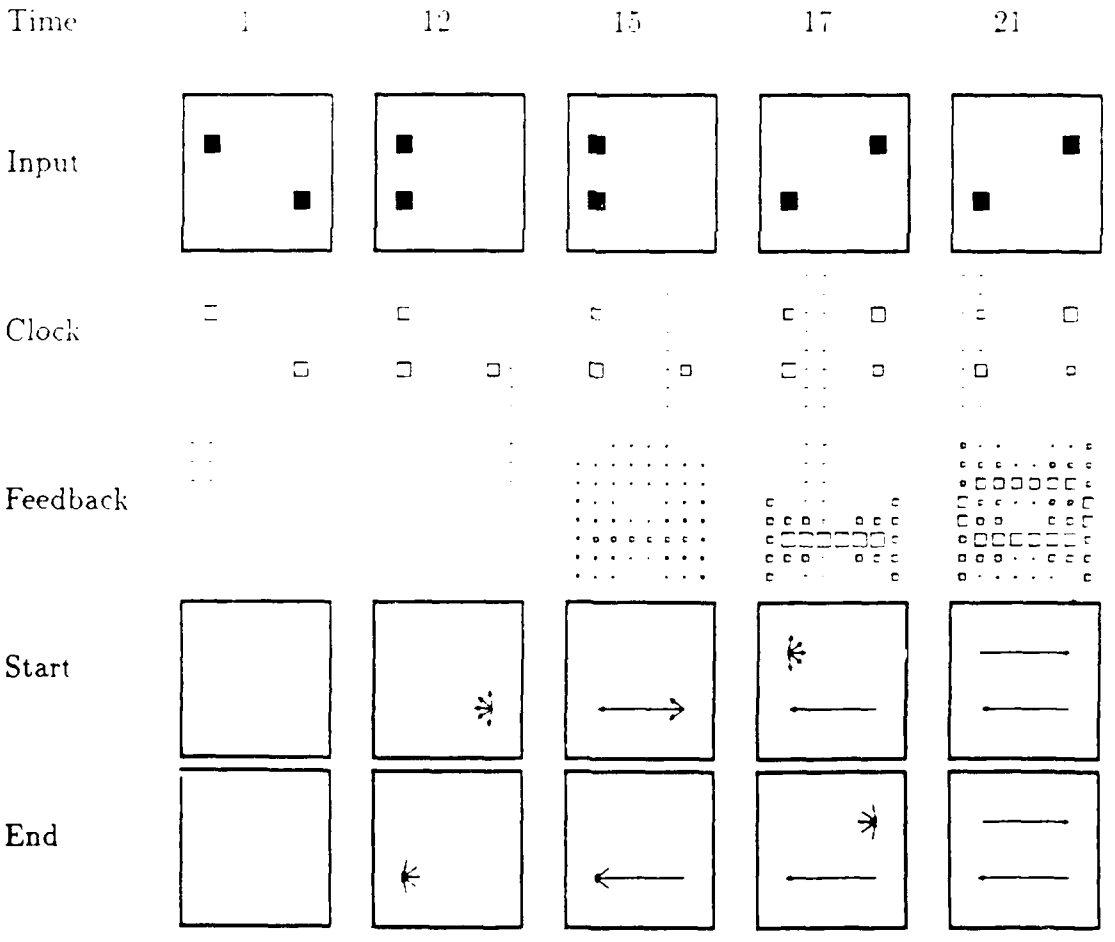

Figure 7.17: Staggered version of the semaphore, showing the effect of negative ISI.

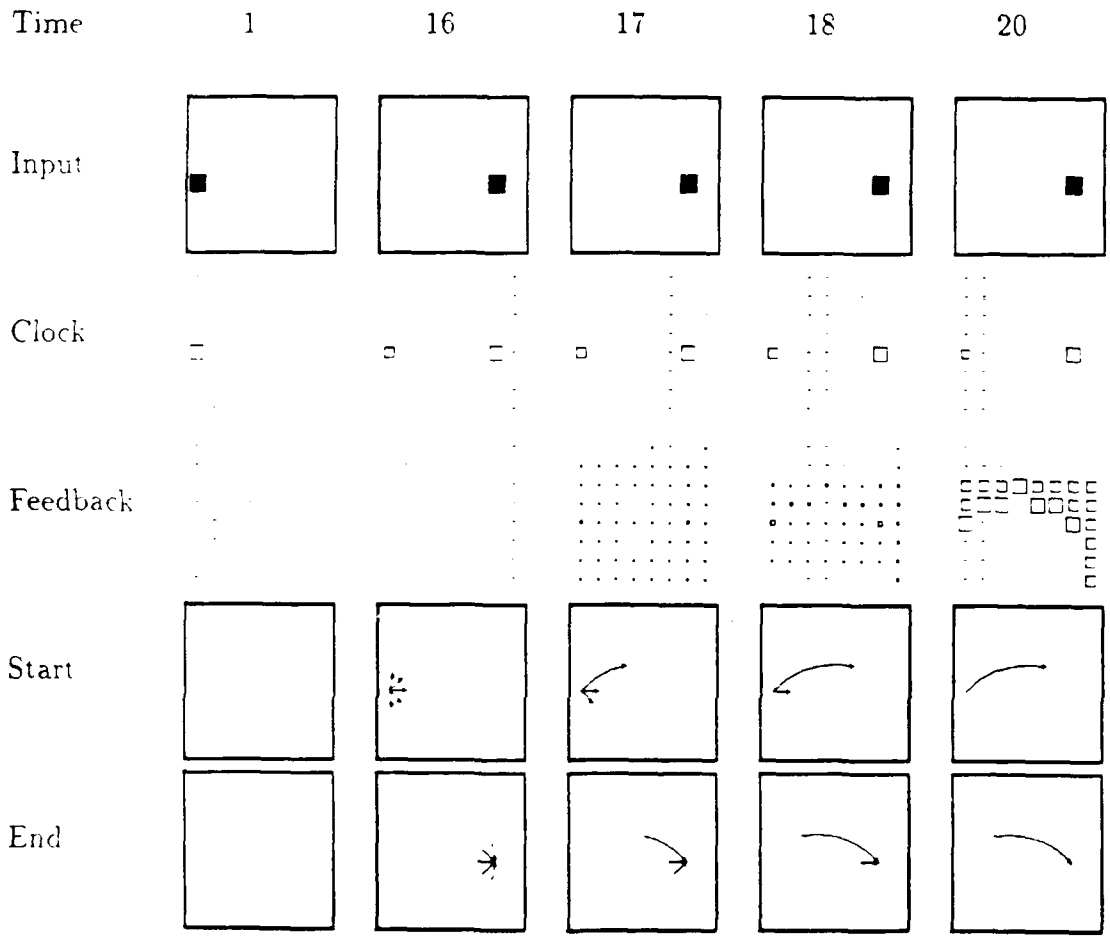

Figure 7.18: Effect of external bias on interpretation of a two-dot stimulus.

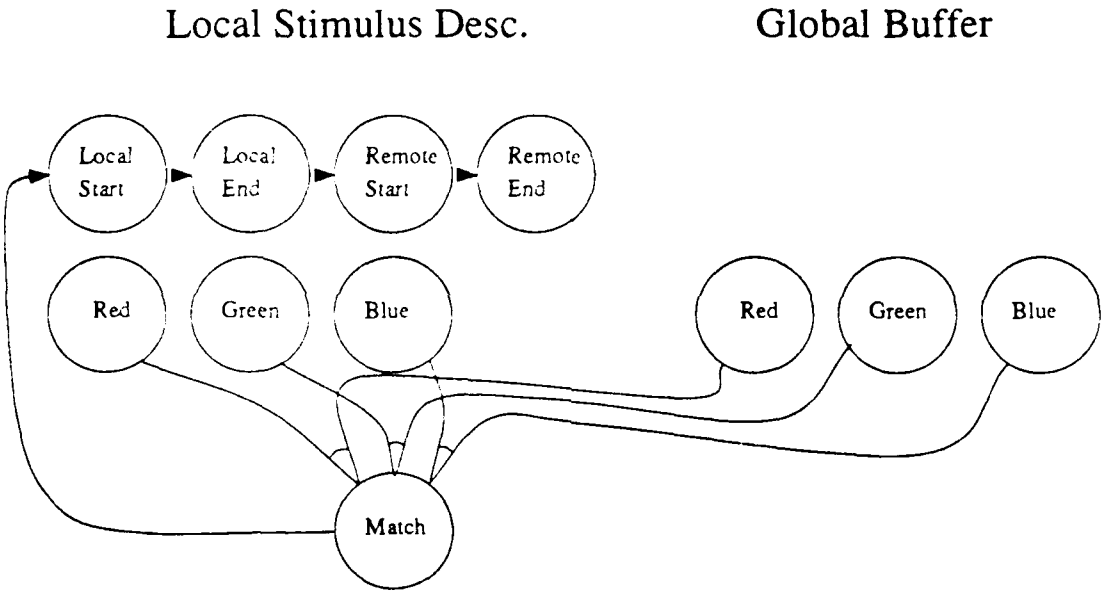

Figure 7.19: Motion network with added mechanism to make use of property match information.

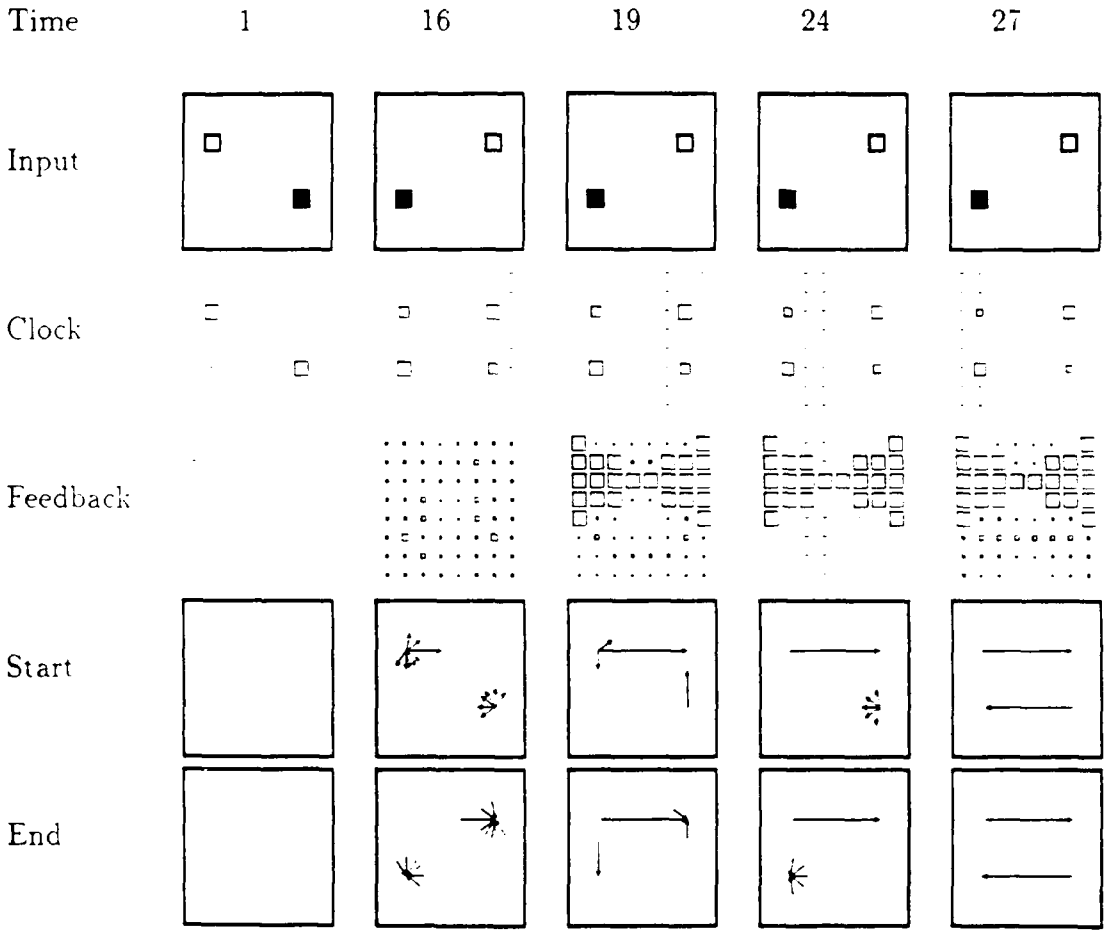

Figure 7.20: Interpretation of a figure with categorical property information.

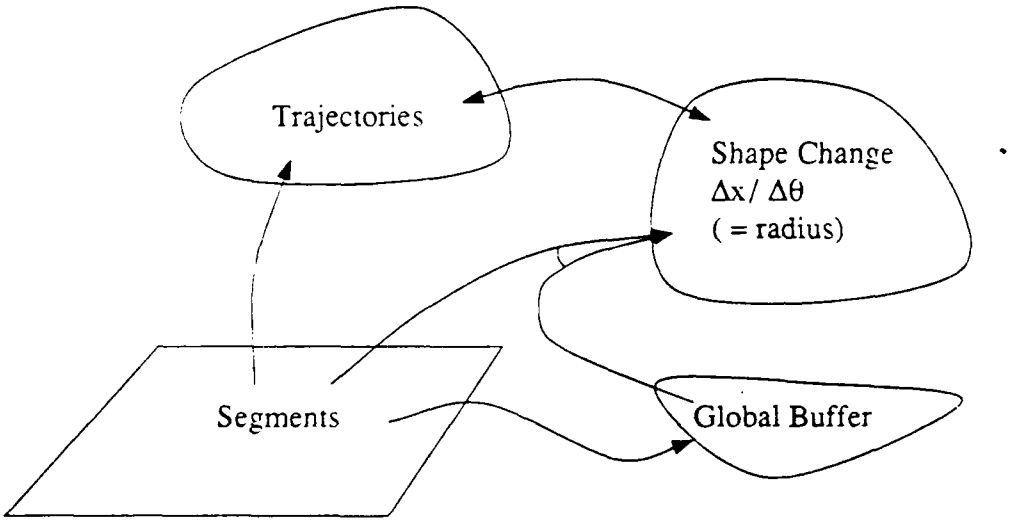

Figure 7.21: Motion network with added mechanism to make use of shape information.

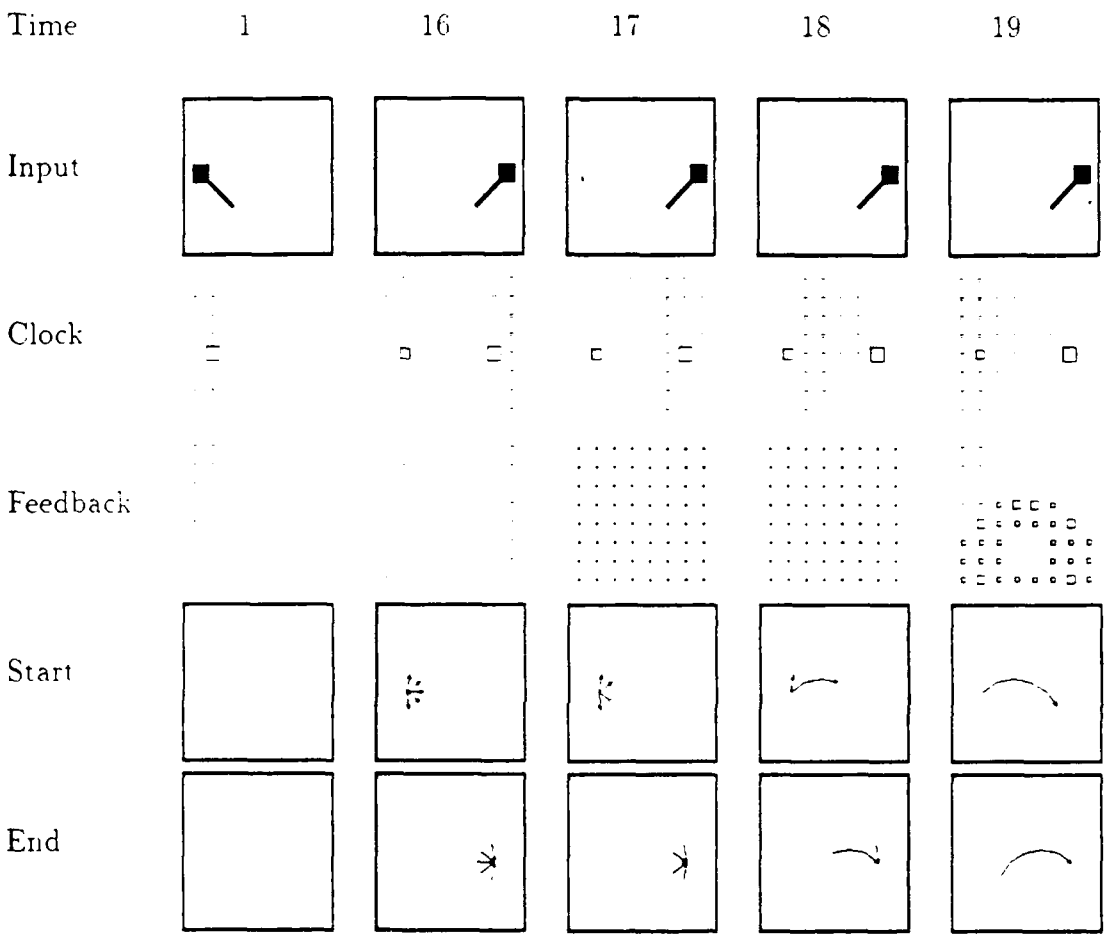

Figure 7.22: Interpretation of a stimulus with shape property information.

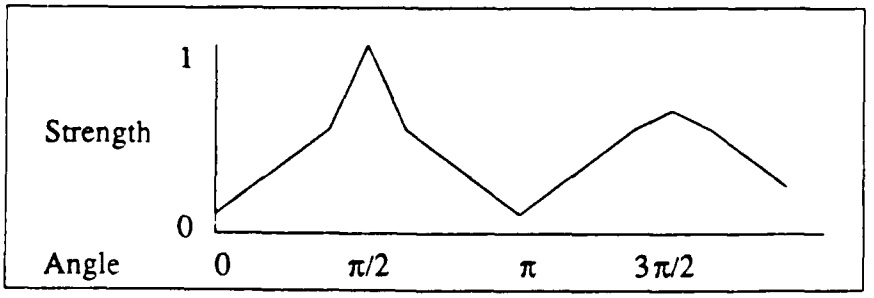

Figure 7.23: Match strength versus angle for the above stimulus.

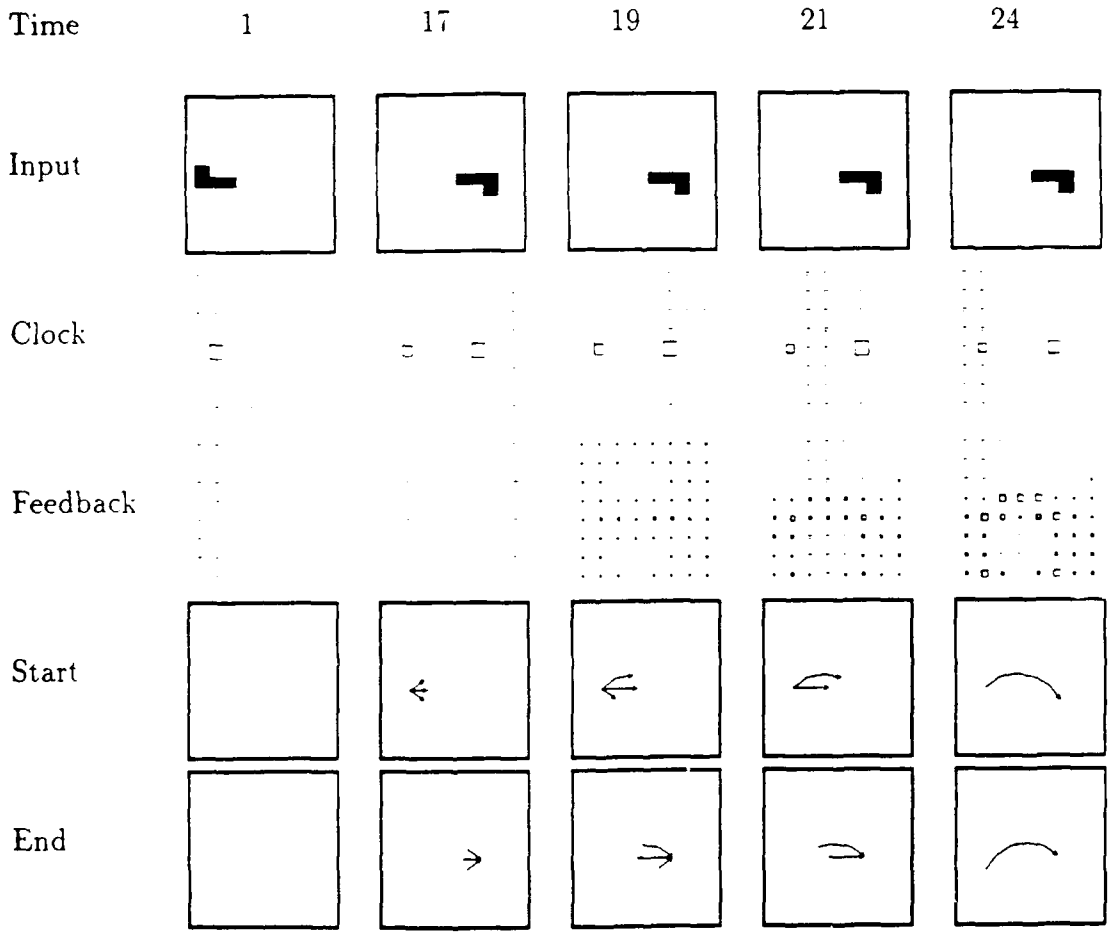

Figure 7.24: Interpretation of a stimulus with weak shape property information at long SOA.

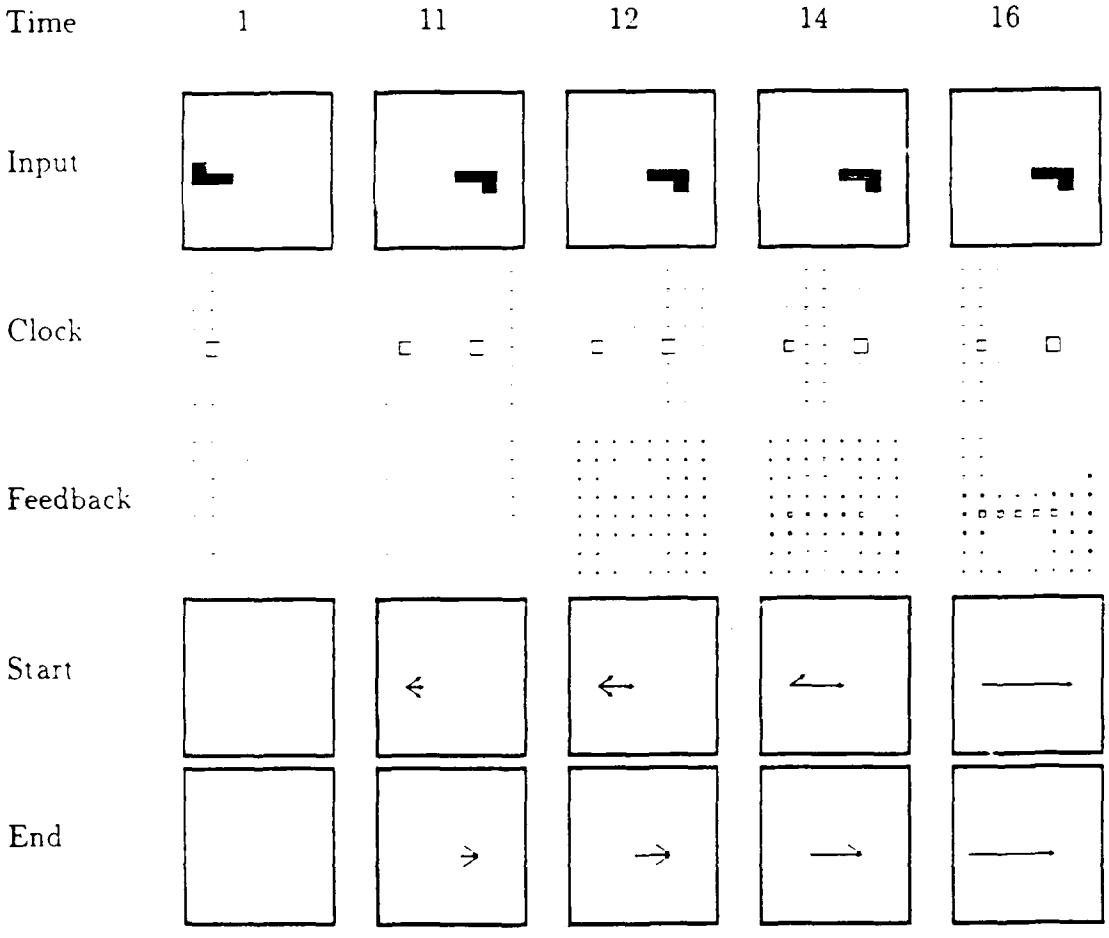

Figure 7.25: Interpretation of a stimulus with weak shape property information at short SOA.

8 Summary, Conclusions, and Future Work

8.1 Summary

The goal of the research described in this thesis is first and foremost to understand human motion perception at the architectural level. That is, we set out to discover what the major subsystems of the motion understanding system are and as much as possible about how they work. The approach taken here is to review the knowledge gathered by vision researchers and to try to construct a systems-level model that is *consistent* with their findings and in addition makes good computational sense.

Chapter Two presents the pieces of the puzzle. Results from psychophysics, neurophysiology and psychology all impose constraints on theories of how motion perception works. Of particular interest is the distinction between apparent and short-range motion. It seems clear that there are two fundamentally different underlying systems, but it is much less obvious what the limits of the two systems are, how much overlap there is, or which psychological effects map to which system. Chapter Three discusses existing models of motion perception drawn from both biological and computer vision perspectives. There are several models of short-range motion that are both biologically and computationally plausible. Existing models of long-range or apparent motion perception are much less satisfactory.

Chapter Four presents a loosely specified architectural model of motion processing that is claimed to account for a broader range of phenomena than any previous model. The ideas presented here were developed in collaboration with Jerry Feldman and Nigel Goddard, and are presented from their own viewpoints in [Feldman, 1988] and [Goddard]. The architecture partitions the motion processing system into three subsystems. The first or low-level subsystem constructs a retinotopic map of motion-like properties of the spatiotemporal contrast distribution. It also may do some integration of local measurements to help resolve the aperture problem. Its measurements are necessarily noisy and uncertain, however. Dealing with this uncertainty is one of the jobs of the next or intermediate-level motion subsystem. In conjunction with an unspecified segmentation process, the intermediate-level system partitions the scene into segments (shape primitives) and assigns them trajectories

from a primitive set. Its computation is heavily driven by the low-level system, but it can also make use of form information, *a priori* knowledge about plausibility of particular trajectories, property match information, categorical knowledge, and other information. It combines all of its inputs to arrive at a non-retinotopic representation of the world in terms of primitive forms moving along primitive trajectories. The last or high-level motion system is concerned with recognition of characteristic motions and motion sequences. Complex events are represented by finite automata or state machines that specify sequences of primitive motions and events. Recognizing and event consists of activating one of the state machines and bringing it into phase with the world events recovered by the intermediate level.

Good models of low-level motion processing exist, and a model of the high level is being developed in [Goddard]. In Chapter Five, therefor, we turn to the intermediate level as the least well understood part of the motion system. The intermediate level computation can be modelled as a generalization of the Hough transform, provided that solutions can be found to the problems of representing the parameter spaces, time, and competition between inconsistent interpretations. The second half of Chapter Five reviews methods for handling large parameter spaces and presents ways of encoding time using predictable temporal response functions. Chapter Six is devoted to a general method of handling competition and other gestalt-like interactions in Hough transform networks. The method, which is called feature binding, allows output parameter space units to compete for ownership of input units without requiring all possible ownership relations to be instantiated. The simplest version can be viewed as an approximation to a well-known continuous relaxation algorithm [Hummel and Zucker, 1983]. The technique can be extended to incorporate external biases and to work with hierarchical and interpolation-coded networks as well.

Finally, in Chapter Seven the techniques developed in Chapters Five and Six are used to construct a connectionist network that interprets motion sequences drawn from a simple domain. The domain consists of presegmented blobs moving along straight or circular paths in two dimensions. Temporal sampling may be arbitrarily coarse, so the domain includes most classical apparent motion stimuli. Design of the network involves taking into account a number of subtle issues, particularly those involving feature binding with distributed parameter space representations and temporally sensitive connections. The network's behavior agrees qualitatively with human interpretations of a large number of apparent motion stimuli. It appears that it could also be extended to handle more complex inputs and to correct problems with the current design.

8.2 Predictions

Since the central claim of this thesis relates to the architecture of motion perception, it is not surprising that most of its predictions do too. This means, unfortunately, that

for the most part they do not have the detailed character that would be of most use to psychophysicists and neurophysiologists. This is not to say that the model cannot be falsified – only that the model is clearer about what the architecture is than about how it is implemented. It is precisely the implementation details, however, that are most accessible to the experimentalist. The space of possible connectionist solutions to the representation problems posed in Chapter Five is large. One particular solution was developed in Chapter Seven, but the intent there was to show feasibility rather than to develop a model that is correct in all its details. The paragraphs that follow will present predictions in decreasing order of generality.

Existence of Trajectories The visual system contains an explicit representation of trajectories. The representation receives and integrates information about motions which are extended in space and time, and is the substrate for apparent motion. Since the space of trajectories is large, its neural representation is likely to be distributed or encoded in some fairly complex way. This may make it difficult to find by current recording techniques. One strong clue would be evidence of facilitory interactions over time intervals of several hundred milliseconds. The trajectory representation should certainly show up psychophysically, and in this context McKee's results are very encouraging.

The fact that trajectory representation is distributed should cause the system to break down due to crosstalk in complex situations. In such cases attentional focus would be needed to interpret the scene. Dick *et al.* [1987] suggested that apparent motion was an attentive process in the Treisman sense; we would attribute their results to crosstalk rather than to any inherently serial component of long-range motion processing.

Existence of Segments Trajectories describe the motions of relatively complex objects such as shape primitives rather than lower level tokens. It is these higher level primitives that move in apparent motion as well. This latter claim could be tested by repeating some of Ullman's experiments (which seemed to show the opposite) using more powerful ways of eliminating short-range motion. For example, one might construct broken-wheel stimuli using random dot kinematograms, random dot stereograms, or isoluminant red-green bars.

Existence of Local Clocks In the network of Chapter Seven, appearance of a new blob triggers a decay clock at its location. Decay clocks may not turn out to be correct, but there should be some local mechanism that allows time since stimulus onset to be measured fairly accurately over 500 to 1000 milliseconds. The representation need not be terribly explicit – for example, it might take the form of decay of the total activity in a local segment descriptor, rather than a dedicated clock unit.

Competition Across Representation Layers In the network design competition between rival interpretations is not direct, but is instead mediated by units in the parameter space of features being interpreted. This idea (*i.e.* some variant of feature binding) offers yet another possible explanation for the top-down connections that found throughout the hierarchy of visual cortex. There are, of course, many other possible explanations.

Velocity Discrimination at Low and Intermediate Levels It is claimed that the low-level system is driven entirely by simple properties of the contrast distribution, in the style of the spatiotemporal frequency models described in Chapter Three. This would imply that 'non-fourier' motion, motion of chromatic patterns, et cetera should be mediated by the intermediate level system. A weak prediction is that these second-order motion phenomena will show Weber fractions for velocity discrimination of around 10%, like those for apparent motion. The recent result of Turano and Pantle [1989] may therefor weaken the argument.

8.3 Future Work

The work described in this thesis can be extended in many different directions. Among these are:

Extending the Implementation As stated at the end of Chapter Seven, there are a number of ways that the network implementation could be modified to handle more phenomena or to repair faults in the design. Of these, extension to incorporate feedback from a higher level seems the most immediately interesting. The idea here would be to use Goddard's network or some simplification of it to recognize characteristic motion sequences, and let priming from the higher level prime the network so that it converges more quickly when given the expected stimulus.

Learning Trajectories It seems likely that a relatively simple form of competitive learning [Rumelhart and Zipser, 1986] would allow the network to learn to represent trajectories that it encounters frequently. This would be of great interest, since it is quite possible that the human trajectory representation is at least partly learned or acquired during development. The problem could be most easily explored by simplifying the problem to one dimension. Suppose that initially the network consists of a set of blob descriptors arranged in a line. Each location also has trajectory Start and End units at each location, but the units connect to remote locations with random static weights and preferred asynchronies. Now suppose that the network is presented with dots moving along trajectories with random lengths at some fixed speed. Feature binding will cause those trajectory units that have connections to the right locations

to compete, even though they cannot be said to represent anything in particular, and one will emerge as the winner. Suppose that the winner increases the static weight of its connection to all locations that were stimulated, and also adjusts the preferred asynchronies of its sites toward whatever asynchrony they actually saw. Losers of the competition would decrease their static weights but leave their preferred asynchronies unchanged. Over time, one would expect the smoothness of the presented stimuli to organize the trajectory unit inputs into spatiotemporally oriented receptive fields like those that were built into the model of Chapter Seven. In addition, the static weights might come to reflect the frequency of co-occurrence of local and remote stimuli, giving rise naturally to nearest neighbor preference.

Improvements to Feature Binding The feature binding technique appears to have applications far beyond motion processing. In order to make it as useful as possible, a number of things must be done. First, the analysis should be broadened to cover some of the extensions described at the end of Chapter Six. It would also be useful to either prove that the technique does in fact converge unconditionally, or to characterize those situations in which it doesn't. Finally, it should be applied to a variety of other problems, so that we can get a feeling for whether it is as useful as it appears to be.

8.4 Conclusion

The work presented here represents an early attempt to understand human motion perception in its entirety. The broad scope of the work has required a corresponding coarseness of detail. However, the intermediate-level motion network of Chapter Seven gives a better account of the phenomena than any previous computational model. Significant contributions have also been made in the area of connectionist technique. We feel that the architectural approach taken here has provided a useful framework for synthesizing results from a variety of disciplines, and hope that it will serve as a foundation on which more detailed models can be built.

Bibliography

[Adelson and Bergen, 1985] Edward H. Adelson and James R. Bergen. "Spatiotemporal Energy Models for the Perception of Motion," *Journal of the Optical Society of America A*, 2(2), 1985.

[Adelson and Bergen, 1986] Edward H. Adelson and James R. Bergen. "The Extraction of Spatio-temporal Energy in Human and Machine Vision." In *IEEE Workshop on Motion Representation and Analysis*, Kiawah, S. C., May 1986.

[Albright, 1984] Thomas D. Albright, "Direction and Orientation Selectivity of Neurons in Visual Area MT of the Macaque," *Journal of Neurophysiology*, 52(6):1106–1130, December 1984.

[Allman *et al.*, 1985] John Allman, Francis Miezin, and EveLynn McGuinness, "Stimulus Specific Responses From Beyond the Classical Receptive Field: Neurophysiological Mechanisms for Local–Global Comparisons in Visual Neurons," *Annual Review of Neuroscience*, 8:407–430, 1985.

[Anstis, 1970] Stuart Anstis, "Phi Movement as a Subtraction Process," *Vision Research*, 10:1411–1430, 1970.

[Anstis and Giaschi, 1985] Stuart Anstis and Deborah Giaschi, "Adaptation to Apparent Motion," *Vision Research*, 25(8):1051–1062, 1985.

[Anstis and Ramachandran, 1986] Stuart Anstis and V. S. Ramachandran, "Entrained Path Deflection in Apparent Motion," *Vision Research*, 26(10):1731–1739, 1986.

[Attneave and Block, 1973] Fred Attneave and Gene Block, "Apparent Movement in Tridimensional Space," *Perception & Psychophysics*, 13(2):301–307, 1973.

[Baker and Braddick, 1985] Curtis L. Baker, Jr. and Oliver J. Braddick, "Eccentricity-dependent Scaling of the Limits for Short-range Apparent Motion Perception," *Vision Research*, 25:803–812, 1985.

[Ballard, 1981] Dana H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, 13:111–122, 1981.

[Ballard, 1984] Dana H. Ballard, "Parameter Networks: Towards a Theory of Low-Level Vision," *Artificial Intelligence*, 22, 1984.

[Ballard, 1986a] Dana H. Ballard, "Cortical Connections and Parallel Processing: Structure and Function," *The Behavioral and Brain Sciences*, 9(1), March 1986.

[Ballard, 1986b] Dana H. Ballard, "Form Perception Using Transformation Networks: Polyhedra," Technical Report 148, University of Rochester Computer Science Department, October 1986.

[Ballard, 1986c] Dana H. Ballard, "Interpolation Coding: A Representation for Numbers in Neural Models," Technical Report 218, University of Rochester Computer Science Department, September 1986.

[Ballard and Sabbah, 1983] Dana H. Ballard and Daniel Sabbah, "Viewer Independent Shape Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(5):653–660, 1983.

[Bandopadhay, 1986] Amit Bandopadhay, *A Computational Study of Rigid Motion Perception*, PhD thesis, University of Rochester, 1986, also published as Computer Science Department TR 221.

[Barlow and Levick, 1965] H. B. Barlow and W. R. Levick, "The Mechanism of Directionally Selective Units in the Rabbit's Retina," *Journal of Physiology*, 178, 1965.

[Baro and Levinson, 1988] John A. Baro and Eugene Levinson, "Apparent Motion Can be Perceived Between Patterns with Dissimilar Spatial Frequencies," *Vision Research*, 28(12):1311–1313, 1988.

[Barrow and Tenenbaum, 1978] H. G. Barrow and J. M. Tenenbaum, "Recovering Intrinsic Scene Characteristics From Images," In Allen R. Hanson and Edward M. Riseman, editors, *Computer Vision Systems*, pages 3–26. Academic Press, Inc., 1978.

[Baylor, 1987] Denis A. Baylor, "Photoreceptor Signals and Vision," *Investigative Ophthalmology and Visual Science*, 28:34–49, January 1987.

[Beverly and Regan, 1973] K. I. Beverly and D. Regan, "Evidence for the Existence of Neural Mechanisms Selectively Sensitive to the Direction of Movement in Space," *Journal of Physiology*, 235:17–29, 1973.

[Beverly and Regan, 1979] K. I. Beverly and D. Regan, "Separable Aftereffects of Changing-size and Motion-in-depth: Different Neural Mechanisms," *Vision Research*, 19:727–732, 1979.

[Biederman, 1987] Irving Biederman, "Recognition by Components: A Theory of Human Image Understanding," *Psychological Review*, 94:115–147, 1987.

[Binford, 1971] Thomas O. Binford, "Visual Perception by Computer," In *IEEE Conference on Systems and Control*, Miami, April 1971.

[Bisti *et al.*, 1985] S. Bisti, G. Carmignoto, L. Galli, and L. Maffei, "Spatial Frequency Characteristics of Neurones of Area 18 in the Cat: Dependence on the Velocity of the Visual Stimulus," *Journal of Physiology*, 359:259–268, 1985.

[Blake and Zisserman, 1987] A. Blake and A. Zisserman, *Visual Reconstruction*, The MIT Press, Cambridge, Massachusetts, 1987.

[Blasdel and Fitzpatrick, 1984] Gary G. Blasdel and David Fitzpatrick, "Physiological Organization of Layer 4 in Macaque Striate Cortex," *Journal of Neuroscience*, 4(3):880–895, March 1984.

[Braddick, 1974] Oliver J. Braddick, "A Short-Range Process in Apparent Motion," *Vision Research*, 14, 1974.

[Braddick, 1980] Oliver J. Braddick, "Low-level and High-level Processes in Apparent Motion," *Philosophical Transactions of the Royal Society of London B*, 290:137–151, 1980.

[Breitmeyer and Ritter, 1986] Bruno G. Breitmeyer and Alysia Ritter, "The Role of Visual Pattern Persistence in Bistable Stroboscopic Motion," *Vision Research*, 26(11):1801–1806, 1986.

[Brooks, 1987] Rodney A. Brooks, "Intelligence Without Representation," In *Proc. Workshop on Foundations of Artificial Intelligence*, pages 1–21, 1987.

[Brown, 1983a] Christopher M. Brown, "Hierarchical Cache Accumulators for Sequential Mode Estimation," Technical Report 125, University of Rochester Computer Science Department, July 1983.

[Brown, 1983b] Christopher M. Brown, "Inherent Bias and Noise in the Hough Transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(5):493–505, September 1983.

[Bundesen *et al.*, 1983] Claus Bundesen, Axel Larsen, and Joyce E. Farrell, "Visual Apparent Movement: Transformations of Size and Orientation," *Perception*, 12:pp 549–558, 1983.

[Burt and Sperling, 1983] Peter Burt and George Sperling. "Time, Distance and Feature Trade-offs in Visual Apparent Motion," *Psychological Review*, 88(2):171–195, 1983.

[Califano *et al.*, 1989] A. Califano, R. M. Bolle, and R. W. Taylor, "Generalized Neighborhoods: A New Approach to Complex'Parameter Feature Extraction," In *Proc. Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, June 1989.

[Cavanagh *et al.*, 1985] Patrick Cavanagh, John Boeglin, and Olga Favreau, "Perception of Motion in Equiluminous Kinematograms," *Perception*, 14:151–162, 1985.

[Cavanagh *et al.*, 1984] Patrick Cavanagh, C. W. Tyler, and Olga Favreau, "Perceived Velocity of Moving Chromatic Gratings," *Journal of the Optical Society of America A*, 1, 1984.

[Chang and Julesz, 1983] J. J. Chang and Bela Julesz, "Displacement Limits for Spatial Frequency Filtered Random-dot Cinematograms in Apparent Motion," *Vision Research*, 23:1379–1386, 1983.

[Chen, 1985] Lin Chen, "Topological Structure in the Perception of Apparent Motion," *Perception*, 14:197–208, 1985.

[Chou, 1988] Paul Bao-Luo Chou, *The Theory and Practice of Bayesian Image Labelling*, PhD thesis, University of Rochester, 1988, also published as Computer Science Department TR 258.

[Cooper, 1989] Paul Cooper, *Parallel Object Recognition From Structure (The Tinkertoy Project)*, PhD thesis, University of Rochester, 1989.

[Derrington and Henning, 1987] Andrew M. Derrington and G. Bruce Henning, "Errors in Direction-of-Motion Discrimination with Complex Stimuli," *Vision Research*, 27, 1987.

[DeYoe and Van Essen, 1985] Edgar A. DeYoe and David C. Van Essen, "Segregation of Efferent Connections and Receptive Field Properties in Visual Area V2 of the Macaque," *Nature*, 317:pp 58–61, September 1985.

[Dick *et al.*, 1987] Miri Dick, Shimon Ullman, and Dov Sagi, "Parallel and Serial Processes in Motion Detection," *Science*, 237:400–402, July 1987.

[Elman, 1988] Jeffrey L. Elman, "Finding Structure in Time," CRL Technical Report 8801, UCSD Center for Research in Language, April 1988.

[Exner, 1875] Sigmund Exner, "Über das Sehen von Bewegungen und die Theorie des zusammengesetzen Auges," *Sitzungsberichte Akademie Wissenschaft Wien*, 72:156–190, 1875.

[Farrell, 1983] Joyce E. Farrell. "Visual Transformations Underlying Apparent Movement." *Perception & Psychophysics*, 33(1):pp 85–92, 1983.

[Farrell and Shepard, 1981] Joyce E. Farrell and Roger N. Shepard, "Shape, Orientation and Apparent Rotational Motion," *Journal of Experimental Psychology, Human Perception and Performance*, 7(2):pp 477–486, 1981.

[Feldman, 1982] Jerome A. Feldman, "Dynamic Connections in Neural Networks." *Biological Cybernetics*, 46:27–39, 1982.

[Feldman, 1985] Jerome A. Feldman, "Four Frames Suffice: a Provisional Model of Vision and Space," *The Behavioral and Brain Sciences*, 8, 1985.

[Feldman, 1988] Jerome A. Feldman, "Time, Space and Form in Vision," Technical Report 244, University of Rochester Computer Science Department, November 1988.

[Feldman and Ballard, 1982] Jerome A. Feldman and Dana Ballard, "Connectionist Models and Their Properties," *Cognitive Science*, 6, 1982.

[Feldman *et al.*, 1988] Jerome A. Feldman, Mark A. Fanty, Nigel H. Goddard, and Kenton J. Lynne, "Computing With Structured Connectionist Networks," *Communications of the Assoc. for Computing Machinery*, 31:170–187, February 1988.

[Fennema and Thompson, 1979] C. L. Fennema and W. B. Thompson, "Velocity Determination in Scenes Containing Several Moving Objects," *Computer Graphics and Image Processing*, 9, 1979.

[Fitzpatrick *et al.*, 1985] D. Fitzpatrick, J. S. Lund, and G. G. Blasdel, "Intrinsic Connections of Macaque Striate Cortex: Afferent and Efferent Connections of Lamina 4c," *Journal of Neuroscience*, 5:3329–3349, 1985.

[Fleet and Jepson, 1984] David J. Fleet and Allan D. Jepson, "A Cascaded Filter Approach to the Construction of Velocity Sensitive Mechanisms," Rsch. in Biological and Computational Vision Technical Report RBCV-TR-84-6, University of Toronto, 1984.

[Foster, 1975] D. H. Foster, "Visual Apparent Motion of Some Preferred Paths in the Rotation Group SO(3)," *Biological Cybernetics*, 18:81–89, 1975.

[Foster *et al.*, 1985] Kent H. Foster, James P. Gaska, Miriam Nagler, and Daniel A. Pollen, "Spatial and Temporal Frequency Selectivity of Neurones in Visual Cortical Areas V1 and V2 of the Macaque Monkey," *Journal of Physiology*, 365:331–363, 1985.

[Friedberg, 1984] Stuart A. Friedberg, "Symmetry Evaluators," Technical Report 134, University of Rochester Computer Science Department, March 1984, (Revised January 1986).

[Geman and Geman, 1984] Stuart Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution and Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):493–505, 1984.

[Giaschi and Anstis, 1989] Deborah Giaschi and Stuart Anstis, "The Less You See It, the Faster It Moves," *Vision Research*, 29(3):335–348, 1989.

[Goddard] Nigel. H. Goddard, *The Representation and Use of Visual Motion*, PhD thesis, University of Rochester, forthcoming.

[Goddard *et al.*, 1988] Nigel H. Goddard, Kenton J. Lynne, and Toby Mintz, "Rochester Connectionist Simulator," Technical Report 233 (revised), University of Rochester Computer Science Department, March 1988.

[Green, 1983] Marc Green, "Inhibition and Facilitation of Apparent Motion by Real Motion," *Vision Research*, 23(9):861–865, 1983.

[Green and Odom, 1986] Marc Green and J. Vernon Odom, "Correspondence Matching in Apparent Motion: Evidence for Three-Dimensional Spatial Representation," *Science*, 233:1427–1429, Sept. 1986.

[Gregory, 1970] R. L. Gregory, *The Intelligent Eye*, McGraw-Hill, New York, 1970.

[Grzywacz and Hildreth, 1987] Norberto M. Grzywacz and Ellen C. Hildreth, "Incremental Rigidity Scheme for Recovering Structure From Motion: Position-based Versus Velocity-based Formulations," *Journal of the Optical Society of America A*, 4(3):503–518, March 1987.

[Heeger, 1986] David Heeger, "Depth and Flow from Motion Energy," In *AAAI-86*, 1986.

[Heeger, 1987] David Heeger, "Optical Flow from Spatiotemporal Filters," In *Proc. 1st Intl. Conf. Computer Vision*, pages 181–190, London, 1987.

[Hildreth, 1984] Ellen C. Hildreth, "The Computation of the Velocity Field," *Proceedings of the Royal Society of London B*, 221, 1984.

[Hildreth and Koch, 1987] Ellen C. Hildreth and Christof Koch, "The Analysis of Visual Motion: From Computational Theory to Neuronal Mechanisms," *Annual Review of Neuroscience*, 10:477–533, 1987.

[Hinton, 1981] Geoffrey E. Hinton. "Shape Representation in Parallel Systems." In *Proc. 7th International Joint Conference on Artificial Intelligence*, pages 1088-1096, Vancouver, 1981.

[Hoffman and Flinchbaugh, 1982] D. D. Hoffman and B. E. Flinchbaugh, "The Interpretation of Biological Motion," *Biological Cybernetics*, 42:195–204, 1982.

[Horn and Schunk, 1981] B. K. P. Horn and B. G. Schunk, "Determining Optical Flow," *Artificial Intelligence*, 17, 1981.

[Hummel and Zucker, 1983] Robert A. Hummel and Steven W. Zucker, "On the Foundations of Relaxation Labelling Processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(3):267–287, 1983.

[Johansson, 1973] Gunnar Johansson, "Visual Perception of Biological Motion and a Model for Its Analysis," *Perception and Psychophysics*, 14:201–211, 1973.

[Johansson, 1976] Gunnar Johansson, "Spatio-temporal Differentiation and Integration in Visual Motion Perception," *Psychological Research*, 38:379–393, 1976.

[Jordan, 1986] Michael I. Jordan. "Serial Order: A Parallel Distributed Processing Approach," Report 8604, UCSD Institute for Cognitive Science, 1986.

[Julesz, 1971] Bela Julesz, *Foundations of Cyclopean Perception*, University of Chicago Press, Chicago, 1971.

[Kearney *et al.*, 1987] J. K. Kearney, W. B. Thompson, and D. L. Boley, "Optical Flow Estimation: an Error Analysis of Gradient-based Methods with Local Optimization," *IEEE PAMI-9*, 9(2), 1987.

[Keck *et al.*, 1976] Max J. Keck, Thomas D. Palella, and Allan Pantle, "Motion Aftereffect as a Function of the Contrast of Sinusoidal Gratings," *Vision Research*, 16:187–191, 1976.

[Kelly and Burbeck, 1987] D. H. Kelly and Christina A. Burbeck, "Further Evidence for a Broadband, Isotropic Mechanism Sensitive to High-velocity Stimuli," *Vision Research*, 27(9):1527–1537, 1987.

[Koch *et al.*, 1986] Christof Koch, Jose Marroquin, and Alan Yuille, "Analog "Neuronal" Networks in Early Vision," *Proc. National Acadamy of Science of the USA*, 83:4263–4267, June 1986.

[Koch *et al.*, 1989] Christof Koch, H. T. Wang, Bimal Mathur, Andrew Hsu, and Humbert Suarez, "Computing Optical Flow in Resistive Networks and in the Primate Visual System," In *Proc. Workshop on Visual Motion*, pages 62–72, Irvine, California, March 1989.

[Koffka, 1935] Kurt Koffka, *Principles of Gestalt Psychology*, Harcourt, Brace and Company, New York, 1935.

[Köhler, 1923] W. Köhler, "Zur Theorie der Stroboskopische Bewegung," *Psychologische Forschung*, 3:397–406, 1923.

[Köhler, 1947] W. Köhler, *Gestalt Psychology*, Liveright, New York, 1947.

[Kolers, 1972] Paul A. Kolers, *Aspects of Motion Perception*, Pergamon Press, New York, 1972.

[Kolers and von Grünau, 1976] Paul A. Kolers and Michael W. von Grünau, "Shape and Color in Apparent Motion," *Vision Research*, 16:329–335, 1976.

[Korte, 1915] A. Korte, "Kinematoskopische Untersuchungen," *Zeitschrift für Psychologie*, 72:194–296, 1915.

[Kozlowski and Cutting, 1977] L. T. Kozlowski and J. E. Cutting, "Recognizing the Sex of Walker From Dynamic Point-light Displays," *Perception and Psychophysics*, 21(6):575–580, 1977.

[Lee and Aronson, 1974] D. L. Lee and E. Aronson, "Visual Proprioceptive Control of Standing in Human Infants," *Perception and Psychophysics*, 15:529–532, 1974.

[Lee and Lishman, 1975] D. L. Lee and R. Lishman, "Visual Proprioceptive Control of Stance," *Journal of Human Movement Studies*, 1:87–95, 1975.

[Lennie, 1980] Peter Lennie, "Parallel Visual Pathways: A Review," *Vision Research*, 20:561–595, 1980.

[Levine, 1985] Martin D. Levine, *Vision in Man and Machine*, McGraw-Hill, New York, 1985.

[Levinson and Sekuler, 1980] Eugene Levinson and Robert Sekuler, "A Two-Dimensional Analysis of Direction-Specific Adaptation," *Vision Research*, 20:103–107, 1980.

[Levinson and Sekuler, 1985] Eugene Levinson and Robert Sekuler, "The Independence of Channels in Human Vision Selective for Direction of Movement," *Journal of Physiology*, 250:347–366, 1985.

[Levitt, 1986] Tod S. Levitt, "Model-based Probabilistic Situation Inference in Hierarchical Hypothesis Spaces," In L. N. Canal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 347–356. Elsevier Science Publishers B. V. (North-Holland), 1986.

[Madden, 1989a] Brian C. Madden. "Apparent Motion, Real Effects." Technical Report 247, University of Rochester Computer Science Department, forthcoming 1989.

[Madden, 1989b] Brian C. Madden, "Space, Time and Apparent Motion," Technical Report 246, University of Rochester Computer Science Department, forthcoming 1989.

[Marr, 1982] David Marr, *Vision*. Freeman, San Francisco, 1982.

[Marr and Ullman, 1981] David Marr and Shimon Ullman, "Directional Selectivity and its Use in Early Visual Processing." *Proceedings of the Royal Society of London B*, 211:151–180, 1981.

[Matthies *et al.*, 1987] Larry Matthies, Richard Szeliski, and Takeo Kanade, "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences." Technical Report CMU-CS-87-185, Computer Science Department, Carnegie Mellon University, December 1987.

[Maunsell, 1987] John H. R. Maunsell, "Physiological Evidence for Two Visual Subsystems." In L. M. Vaina, editor, *Matters of Intelligence*, pages 59–87. D. Reidel Publishing Company, Dordecht, 1987.

[Maunsell *et al.*, 1989] John H. R. Maunsell, Derryl D. DePriest, and Tara A. Nealy, "The Middle Temporal Visual Area Receives Excitatory Drive Primarily Via the Magnocellular Layers of the LGN," *Investigative Ophthalmology and Visual Science*, 30(3):427, March 1989, (ARVO Supplement).

[Maunsell and Newsome, 1987] John H. R. Maunsell and William T. Newsome, "Visual Processing in Monkey Extrastriate Cortex," *Annual Review of Neuroscience*, 10:363–401, 1987.

[Maunsell and Van Essen, 1983] John H. R. Maunsell and David C. Van Essen, "Functional Properties of Neurons in Middle Temporal Visual Area of the Macaque Monkey. II. Binocular Interactions and Sensitivity to Binocular Disparity," *Journal of Neuroscience*, 49(5):1148–1167, May 1983.

[McKee, 1981] Suzanne P. McKee, "A Local Mechanism for Differential Velocity Detection," *Vision Research*, 21:491–500, 1981.

[McKee and Nakayama, 1988] Suzanne P. McKee and Kenneth Nakayama, "Velocity Integration Along the Trajectory," *Investigative Ophthalmology and Visual Science*, 29:266, 1988, (ARVO Supplement).

[McKee *et al.*, 1986] Suzanne P. McKee, Gerald H. Silverman, and Kenneth Nakayama, "Precise Velocity Discrimination Despite Random Variations in Temporal Frequency and Contrast," *Vision Research*, 26(4):609–619, 1986.

[McKee and Welch, 1989] Suzanne P. McKee and Leslie Welch, "Is there a Constancy for Velocity?," *Vision Research*, 29(5):553–561, 1989.

[McKee and Welch, 1985] Suzanne P. McKee and Leslie Welch, "Sequential Recruitment in the Discrimination of Velocity," *Journal of the Optical Society of America A*, 2(2):243–251, February 1985.

[McLeod *et al.*, 1988] P. McLeod, J. Driver, and J. Crisp, "Visual Search for a Conjunction of Movement and Form is Parallel," *Nature*, 332:154–155, 1988.

[Mikami *et al.*, 1986] Akichika Mikami, William T. Newsome, and Robert H. Wurtz, "Motion Selectivity in Macaque Visual Cortex. II. Spatiotemporal Range of Directional Interactions in MT and V1," *Journal of Neurophysiology*, 55(6):1328–1339, June 1986.

[Mohammed *et al.*, 1983] John L. Mohammed, Robert A. Hummel, and Steven W. Zucker, "A Gradient Projection Algorithm for Relaxation Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(3):330–332, 1983.

[Motter and Mountcastle, 1981] B. C. Motter and V. B. Mountcastle, "The Functional Properties of the Light-Sensitive Neurons of the Posterior Parietal Cortex Studied in Waking Monkeys: Foveal Sparing and Opponent Vector Organization," *Journal of Neuroscience*, 1:3–26, 1981.

[Movshon *et al.*, 1985] J. Anthony Movshon, Edward H. Adelson, Martin S. Gizzi, and William T. Newsome, "The Analysis of Moving Visual Patterns," In C. Chagas R. Gattass C. Gross, editor, *Experimental Brain Research Supplement 11*, pages 117–150. Springer-Verlag, New York, 1985.

[Murray and Buxton, 1987] David W. Murray and Bernard F. Buxton, "Scene Segmentation from Visual Motion Using Global Optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):220–228, March 1987.

[Mutch *et al.*, 1983] Kathleen Mutch, Isabel M. Smith, and Albert Yonas, "The Effect of Two-Dimensional and Three-Dimensional Distance on Apparent Motion," *Perception*, 12:pp 305–312, 1983.

[Nakayama, 1985] Ken Nakayama, "Biological Image Motion Processing: A Review," *Vision Research*, 25(5), 1985.

[Neuhaus, 1930] W. Neuhaus, "Experimentelle Untersuchung der Scheinbewegung," *Archiv für die gesamte Psychologie*, 75:315–458, 1930.

[Newsome *et al.*, 1983] William T. Newsome, Martin S. Gizzi, and J. Anthony Movshon, "Spatial and Temporal Properties of Neurons in Macaque MT," *Investigative Ophthalmology and Visual Science*, 24:106, 1983.

[Newsome *et al.*, 1985] William T. Newsome, Robert H. Wurtz, Max R. Dürsteler, and Akichika Mikami. "Deficits in Visual Motion Processing Following Ibotenic Acid Lesions of the Middle Temporal Visual Area of the Macaque Monkey," *The Journal of Neuroscience*, 5(3):825–840, March 1985.

[Pantle, 1974] Allan Pantle, "Motion Aftereffect Magnitude as a Measure of the Spatio-Temporal Response Properties of Direction-Sensitive Analyzers," *Vision Research*, 14, 1974.

[Pasternak *et al.*] Tatiana Pasternak, Kris Horn, and John H. R. Maunsell, "Deficits in Speed Discrimination Following Lesions of the Lateral Suprasylvian Cortex in the Cat," accepted for publication in *Visual Neuroscience*, date not set.

[Pentland, 1986] Alex P. Pentland, "Perceptual Organization and the Representation of Natural Form," *Artificial Intelligence*, 28(2):493–505, 1986.

[Pentland, 1988] Alex P. Pentland, "Automatic Extraction of Deformable Part Models," Vision Sciences Technical Report 104, MIT Media Lab, July 1988.

[Petersik and Pantle, 1979] J. T. Petersik and A. J. Pantle, "Factors Controlling the Competing Sensations Produced by a Bistable Stroboscopic Motion Display," *Vision Research*, 19:143–154, 1979.

[Petersik *et al.*, 1983] J. Timothy Petersik, Randall Pufahl, and Elizabeth Krasnoff, "Failure to Find an Absolute Retinal Limit of a Putative Short-Range Process in Apparent Motion," *Vision Research*, 23(12):1663–1670, 1983.

[Prazdny, 1986a] K. Prazdny, "Three-dimensional Structure From Long-range Apparent Motion," *Perception*, 15:619–625, 1986.

[Prazdny, 1986b] K. Prazdny, "What variables control (long-range) apparent motion?," *Perception*, 15:37–40, 1986.

[Ramachandran, 1988] V. S. Ramachandran, "Perceiving Shape from Shading," *Scientific American*, 259:76–83, 1988.

[Ramachandran and Anstis, 1983a] V. S. Ramachandran and Stuart M. Anstis, "Displacement Thresholds for Coherent Apparent Motion in Random Dot Patterns," *Vision Research*, 23:1719–1724, 1983.

[Ramachandran and Anstis, 1983b] V. S. Ramachandran and Stuart M. Anstis, "Perceptual Organization in Moving Patterns," *Nature*, 304:529–531, August 1983.

[Ramachandran and Anstis, 1986] V. S. Ramachandran and Stuart M. Anstis, "Figure-ground Segregation Modulates Apparent Motion," *Vision Research*, 26:1969–1975, 1986.

[Ramachandran and Gregory, 1978] V. S. Ramachandran and R. L. Gregory, "Does Color Provide an Input to Human Motion Perception?," *Nature*, 275, 1978.

[Ramachandran *et al.*, 1986] V. S. Ramachandran, V. Inada, and G. Kiama, "Perception of Illusory Occlusion in Apparent Motion," *Vision Research*, 26(10):1741–1749, 1986.

[Ramachandran *et al.*, 1973] V. S. Ramachandran, V. R. Madhusudhan, and T. R. Vidyasagar, "Apparent Movement With Subjective Contours," *Vision Research*, 13:1399–1401, 1973.

[Rashid, 1980] Richard R. Rashid, *Lights: A System for the Interpretation of Moving Light Displays*, PhD thesis, University of Rochester, 1980.

[Regan and Beverly, 1985] D. Regan and K. I. Beverly, "Visual Responses to Vorticity and the Neural Analysis of Optic Flow," *Journal of the Optical Society of America A*, 2(2):280–283, February 1985.

[Reichardt, 1961] Werner Reichardt, "Autocorrelation, a Principle for the Evaluation of Sensory Information by the Central Nervous System," In W. A. Rosenblith, editor, *Sensory Communication*, Wiley, New York, 1961.

[Rock, 1983] Irwin Rock, *The Logic of Perception*, MIT Press, Cambridge, 1983.

[Rock and Ebenholtz, 1962] Irwin Rock and S. Ebenholtz, "Stroboscopic Movement Based on Change of Phenomenal Rather than Retinal Location," *The American Journal of Psychology*, 75:193–207, 1962.

[Rumelhart and McClelland, 1986a] David E. Rumelhart and James L. McClelland, "On Learning the Past Tenses of English Verbs," In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Volume 2: Psychological and Biological Models*, MIT Press, Cambridge, 1986.

[Rumelhart and McClelland, 1986b] David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Volume 1: Foundations*, MIT Press, Cambridge, 1986.

[Rumelhart *et al.*, 1986] David E. Rumelhart, Paul Smolensky, James L. McClelland, and Geoffrey Hinton, "Schemata and Sequential Thought Processes in PDP Models," In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Volume 2: Psychological and Biological Models*, MIT Press, Cambridge, 1986.

[Rumelhart and Zipser, 1986] David E. Rumelhart and David Zipser, "Feature Discovery by Competitive Learning." In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Volume 1: Foundations.* MIT Press, Cambridge, 1986.

[Saito *et al.*, 1986] Hideaki Saito, Masao Yukie, Keiji Tanaka, Kazuo Hikosaka, Yoshiro Fukada, and Eiichi Iwai, "Integration of Direction Signals of Image Motion in the Superior Temporal Sulcus of the Macaque Monkey," *The Journal of Neuroscience*, 6(1):145–157, January 1986.

[Sekuler and Ganz, 1963] Robert William Sekuler and Leo Ganz, "Aftereffect of Seen Motion with a Stabilized Retinal Image," *Science*, 139:419–420, February 1963.

[Shastri, 1985] Lokendra Shastri, *Evidential Reasoning in Semantic Networks: A Formal Theory and its Parallel Implementation*, PhD thesis, University of Rochester, 1985, also published as Computer Science Department TR 166.

[Shepard and Zare, 1982] Roger N. Shepard and Susan L. Zare, "Path-Guided Apparent Motion," *Science*, 220.pp 632–634, 1982.

[Shipp and Zeki, 1985] S. Shipp and S. Zeki, "Segregation of Pathways Leading from Area V2 to Areas V4 and V5 of Macaque Monkey Visual Cortex," *Nature*, 315:pp 322–325, May 1985.

[Smolensky, 1986] Paul Smolensky, "Foundations of Harmony Theory: Cognitive Dynamical Systems and the Subsymbolic Theory of Information Processing." In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Volume 1: Foundations.* MIT Press, Cambridge, 1986.

[Terzopoulos, 1986] Demetri Terzopoulos, "Regularization of Inverse Visual Problems Involving Discontinuities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:413–424, 1986.

[Thompson, 1984] Peter Thompson, "The Coding of Velocity of Movement in the Human Visual System," *Vision Research*, 24(1):41–45, 1984.

[Thompson, 1989] William B. Thompson, "Introduction to the Special Issue on Visual Motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):449–450, May 1989.

[Tiana *et al.*, 1989] Carlo Tiana, Peter Lennie, and Michael D'Zmura, "Parallel Search for Color/Shape and Color/Motion Conjunctions," *Investigative Ophthalmology and Visual Science*, 30(3):160, March 1989, (ARVO Supplement).

[Tolhurst, 1973] D. J. Tolhurst, "Separate Channels for the Analysis of the Shape and the Movement of a Moving Visual Stimulus," *Journal of Physiology*, 231, 1973.

[Torre and Poggio, 1978] V. Torre and Tomaso Poggio, "A Synaptic Mechanism Possibly Underlying Directional Selectivity to Motion," *Proceedings of the Royal Society of London B*, 202:409–416, 1978.

[Triesman, 1985] Anne Triesman, "Preattentive Processing in Vision," *Computer Vision, Graphics, and Image Processing*, 31:156–177, 1985.

[Tsotsos, 1987] John K. Tsotsos, "A 'Complexity Level' Analysis of Vision," In *Proc. First International Conference on Computer Vision*, pages 825–834, London, June 1987.

[Turano and Pantle, 1989] Kathleen Turano and Allan Pantle, "On the Mechanism that Encodes the Movement of Contrast Variations: Velocity Discrimination," *Vision Research*, 29(2):207–222, 1989.

[Ullman, 1979] Shimon Ullman, *The Interpretation of Visual Motion*, MIT Press, Cambridge, Massachussetts, 1979.

[Ullman, 1984] Shimon Ullman, "Maximizing Rigidity: The Incremental Recovery of 3-D Structure From Rigid and Rubbery Motion," *Perception*, 13:255–274, 1984.

[Ullman, 1985] Shimon Ullman, "Visual Routines," In Steven Pinker, editor, *Visual Cognition*, MIT Press, Cambridge, 1985, Reprinted from *Cognition* volume 18, special issue on visual cognition.

[Ungerleider and Mishkin, 1982] L. G. Ungerleider and M. Mishkin, "Two Cortical Visual Systems," In D. J. Ingle, M. A. Goodale, and R. J. W. Mansfield, editors, *Analysis of Visual Behavior*, pages 549–580. MIT Press, Cambridge, Mass., 1982.

[Van Essen, 1985] David C. Van Essen, "Functional Organization of Primate Visual Cortex," In Alan Peters and Edward G. Jones, editors, *Cerebral Cortex, Volume 3: Visual Cortex*. Plenum Press, New York, 1985.

[van Santen and Sperling, 1985] J. P. H. van Santen and George Sperling, "Elaborated Reichardt Detectors," *Journal of the Optical Society of America A*, 2(2), 1985.

[Verri and Poggio, 1987] Alessandro Verri and Tomaso Poggio, "Qualitative Information in the Optical Flow," In *Proc. DARPA Image Understanding Workshop*, pages 825–834, Los Angeles, February 1987.

[von Grünau, 1979a] Michael W. von Grünau, "Form Information is Necessary for the Perception of Motion," *Vision Research*, 19:339–341, 1979.

[von Grünau, 1979b] Michael W. von Grünau, "The Involvement of Illusory Contours in Stroboscopic Motion," *Perception & Psychophysics*, 25(3):205–208, 1979.

[Wallach and O'Connell, 1953] Hans Wallach and D. N. O'Connell, "The Kinetic Depth Effect," *The Journal of Experimental Psychology*, 45:205–217, 1953.

[Watson, 1986] Andrew B. Watson, "Apparent Motion Occurs Only Between Similar Spatial Frequencies," *Vision Research*, 26(10):1727–1730, 1986.

[Watson and Ahumada, 1985] Andrew B. Watson and Albert J. Ahumada, Jr., "Model of Human Visual-Motion Sensing," *Journal of the Optical Society of America A*, 2(2), February 1985.

[Watson *et al.*, 1985] Andrew B. Watson, Albert J. Ahumada, Jr., and Joyce E. Farrell, "Window of Visibility: A Psychophysical Theory of Fidelity in Time-Sampled Visual Motion Displays," *Journal of the Optical Society of America A*, 3(3), March 1985.

[Watson *et al.*, 1980] Andrew B. Watson, Peter G. Thompson, Brian J. Murphy, and Jacob Nachmias, "Summation and Discrimination of Gratings Moving in Opposite Directions," *Vision Research*, 20:341–347, 1980.

[Waxman *et al.*, 1988] Allen M. Waxman, Michael Siebert, R. Cunningham, and Jian Wu, "The Neural Analog Diffusion-Enhancement Layer (NADEL) and early visual processing," In *Proc. SPIE Conf. on Visual Communications and Image Processing 88*, pages 1093–1102, Cambridge, Mass., 1988.

[Waxman *et al.*, 1989] Allen M. Waxman, Jian Wu, and Michael Siebert, "Computing Visual Motion in the Short and the Long: From Receptive Fields to Neural Networks," In *Proc. Workshop on Visual Motion*, pages 156–164, Irvine, California, 1989.

[Wertheimer, 1912] Max Wertheimer, "Experimentelle Studien über das Sehen von Bewegung," *Zeitschrift für Psychologie*, 61:161–265, 1912.

[Witkin *et al.*, 1987] Andrew Witkin, Demetri Terzopoulos, and Michael Kass, "Signal Matching Through Scale Space," *International Journal of Computer Vision*, 1:133–144, 1987.

[Yuille and Grzywacz, 1988] Alan L. Yuille and Norberto M. Grzywacz, "The Motion Coherence Theory," In *Proceedings of the Second Int'l Conf. on Computer Vision*, pages 344–353, Miami, 1988.

[Zihl *et al.*, 1983] J. Zihl, D. von Cramon, and N. Mai, "Selective Disturbance of Movement Vision After Bilateral Brain Damage," *Brain*, 106:313–340, 1983.